Evaluating winning team characteristics for predicting outcomes of DoTa 2 Brackets

Neil Sengupta

Abstract

This paper analyses DoTa 2 multiplayer team matchups and tries to evaluate what team characteristics and matchups between heroes in teams are essential to bring positive outcomes for a team in a match. The paper tries to lay emphasis on not just opposing team lineups, but uses stepwise regression to select and evaluate other salient features of gameplay (like item builds for each player and hero combos).

1 Introduction

DoTa 2 is a free to play multiplayer online battle arena with over a 13 million monthly player base. It is a sequel to the Defense of the Ancients and is most widely known for its intense game strategies amongst professional teams and acute twists in matchups brought about by the interplay of several different gameplay attributes. Essentially, there are two teams of 5 players each. The objective of each team is to destroy the opponent team's base. In this process, each of 5 players control a hero, who have base and special skills, to battle and defeat heroes in the opposing team. For a team to have a successful game, there are several factors that come into play. Firstly, the competitive rating of each individual player is a strong indicator of success in a game. However, a more important feature is the level of proficiency a player has while using a select hero. In professional matchups, hero picks can be randomized or include blocks imposed on certain heroes that a player is known to be good at. Within teams, certain heroes pair well with other hero picks and complement each other's spells. There are also certain restraints. However, picking a good hero combination is vital in determining how the game is played and can sometimes make it easier to reach end game faster.

2 Dataset collection

Valve has an open source Web API. A python wrapper called dota2api is used in order to retrieve data from matches. Some useful responses received using the API call include match history, league listings, live league games, player summaries, team information, heroes picked and item builds over the game and checkpoints conquered by each team.

First, a lot of game match data was collected since it was central to analyzing game performance. There are different game modes that involve game modes for lower ranked players too. However, for the purpose of evaluating brackets for players performing at the highest level, this data was filtered out. The game types that are analyzed in this paper are All Pick, Ranked All Pick, Captain's Draft, Random Draft. Also, invalid match data caused by unexpected player dropouts or connectivity errors were filtered out too. Another constraint applied to the data is that only matches over 1800 seconds are taken into account. Most professional matches last at least 40-45 minutes on average, and times of 30 minutes or less indicate huge early game mistakes and possibly uncharacteristic, incompetent playing strategies by the losing teams.

Next, a lot of individual player data was collected. The player data is more condensed since there are several different player attributes. Each game has ten different players and thus the API served useful to pull relevant match data like Kills, Death, Assists, Item Builds, XP gained/farmed, XP fed to opponent team. For each player, about last 100 games played on competition servers (ESL, Starladder etc.) were used for assessment.

The in game client for DotA2 has a match making mechanism which makes matches for players that are on nearly the same competitive level. In the case of bracketed competitions, the draft ensures a fair competition level for a matchup. The assumption thus made is that there is a fair competition level. Also, an assumption is made that every player has a good grasp over almost every hero they are playing (but not necessarily every hero combination). Although, there are 112 playable heroes in the game, but assuming a level of proficiency amongst the professional players allows the problem to simplified down.

3 Data Analysis

Features and Preprocessing: First the average hero statistics for the games were normalized to have mean 0 and variance 1. Then PCA was run on the statistics to get the composite score for each hero that are stored in vectors $\vec{x}$. This combined together with hero selection data is used to build the feature vector.

$$\Phi(x) = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{112} \\ x_{113} \\ x_{114} \\ \vdots \\ x_{224} \end{bmatrix}$$

In the matrix the first half of the maxtrix $(X_1 \cdots X_{112})$ represents players that are part of the Radiant side and $(X_{113} \cdots X_{224})$ represents players that are part of the Dire side.
For $(X_1 \cdots X_{112})$:

$$\begin{cases} 1, \text{player of radiant plays hero } i \\ 0, \text{no player of radiant plays hero } i \end{cases}$$

For $(X_{113} \cdots X_{224})$: $X_i =$

$$\begin{cases} 1, \text{player of dire plays hero } i \\ 0, \text{no player of dire plays hero } i \end{cases}$$

The outcome of the match is then $Y_j =$

$$\begin{cases} 1, \text{Radiant victory} \\ 0, \text{Dire victory} \end{cases}$$

Logistic regression and then K-fold cross validation was used to minimize the cost function:

$$min_{w,c} \frac{1}{2} w^T w + C \sum_{i=1}^{m} log(exp(-Y_i(\phi(x)_i^T w + c)) + 1)$$

About 5000 match data was collected and then divided into 10 folds. So we 4500 match data for training and the rest as the validation set. When computing the winner of the game, the side with a higher probability was chosen. The computation was done in sci-kit's logistic regression class.

Fold	1	2	3	4	5	6	7	8	9	10
Error(Train)	0.28	0.29	0.3	0.27	0.29	0.25	0.23	0.29	0.33	0.31
Error(Test)	0.47	0.43	0.49	0.42	0.48	0.47	0.42	0.46	0.43	0.46

In this case the training error was about 28% and the testing error was about 45%. This basically entailed that more data about hero combination, item collection, character builds are more important and decisive when predicting the outcome of the match. Using logistic regression, the strength of certain heroes individually, in a team and in match-ups were identified. Adding these into our feature selection would prove to be more useful in classifying and predicting winner of matches.

So, about 60 different inter-hero combinations and match-ups were added into the feature vector and then logistic regression was run to get the following results:

Fold	1	2	3	4	5	6	7	8	9	10
Error(Train)	0.25	0.26	0.23	0.28	0.25	0.25	0.24	0.27	0.29	0.28
Error(Test)	0.4	0.41	0.45	0.43	0.41	0.39	0.41	0.43	0.4	0.42

So, in this case after adding hero combinations the training and testing error were significantly reduced from our earlier case.

As, was seen the inter-hero combination is definitely very vital in predicting outcomes of games. However, considering the vast number of heroes, inter-hero combinations and then combination of features of each hero, a stepwise regression model is proposed. So, basically we start out with no features, then visit a feature of each hero, add it to our feature vector, and then if it increases the model, then we add it to our feature set permanently.

Algorithm 1: Stepwise Regression for adding features

function addFeatures ;
Input : None
Output: Set of Features
numOfFeatures = $\{\}$
featuresNeeded = $\{\}$
while (numOfFeatures $\neq$ featuresNeeded)
Fit a model using the $featuresNeeded$
Calculate $\frac{\partial L(\beta)}{\partial \beta} = \sum_{i=1}^{N} x_i(y_i - P(x_i; B))$
F_{new} = arg max($|\frac{\partial L(\beta)}{\partial \beta}|_i$)
Add feature corresponding to F_{new} to featuresNeeded
numOfFeatures+=1
end
return featuresNeeded

Using the above algorithm we can predict and select the important features of each heroes and thus, the important inter-hero combinations vital for a team's victory. Also, we need to decide the minimum or maximum number of features we need to add to the feature set for each hero. So, we basically pick different number of features and then observe the training and testing error on the model picked above. It was observed that at first the error is higher for lower features, and then the error hits a minimum around 60-70 features.

4 Possible Future Work

In this paper, and attempt was made to take into account hero combination, features and lineups for each hero in order to predict outcome. Possible work could include the personalized level of playing for the individual controlling the game and schematics between individuals who actually play the game and how it could affect the game outcome. Also, other models probably exisiting for more accurate predictions. For example, maybe we could train the set on a convolutional neural network to recognize important features over different layers.

5 References

1)Learning Dota 2 Team Compositions by Atish Agarwala and Michael Pearce
2)http://kevintechnology.com/post/71621133663/using-machine-learning-to-recommend-heroes-for
3)Applications of Machine Learning in Dota 2:Literature Review and Practical Knowledge Sharing
4)Skill-Based Differences in Spatio-Temporal Team Behaviour in Defence of The Ancients 2 (DotA 2)