

Progress report

Sizhe Yuen

November 12, 2018

CS5199 - Individual Masters Project John Thomson

1 Introduction

The parsing of features . Furthermore, most of the steps are scripted to run automatically, such as automatically downloading replays from the OpenDota API, parsing the replays and saving the parsed data. This makes it trivial to get new data as it only takes time, but no manual effort.

2 Machine learning features

Currently, there are two subsets of features used for machine learning: mouse movement and game-specific features. The mouse movement features looks only at the way the player moves their cursor, while the game-specific features look at statistics such as gold per minute and number of attack commands sent.

2.1 Mouse movement

The mouse movement features were extracted following a paper [**mouse-dynamics**] on user identification via mouse dynamics. The paper's methodology split the mouse movements into different types of actions, such as movement followed by left click, movement followed by drag etc. The concept of lower and higher level actions is also defined by the paper. Higher level actions are made up of low level actions. For example, a single click and a mouse movement sequence is defined as a low level action, whereas a sequence of mouse movements followed by a click is defined as a high level action. This approach was adapted but not completely followed due to the different data that is available from parsing replays.

While multiple levels of actions are defined in the paper, only two are defined in this project: **Level 1 actions** and **Level 3 actions**.

Four **level 1 actions** are defined, they are:

- Mouse movement sequence (MM)
- Attack command (AC)
- Move command (MC)
- Spell cast command (CC)

A mouse movement is defined as a sequence of positions of the cursor. Rather than using a fixed interval of time in which the sequence must fit, a threshold is used to end the sequence if no change in cursor position has occurred. This more naturally records a sequence of mouse movements and doesn't break up one sequence of movements to fit within a fixed time interval.

The combination of a MM action followed by a command action defines the **level 3 actions**:

- Mouse movement sequence followed by an attack command (MMAC)
- Mouse movement sequence followed by a move command (MMMC)
- Mouse movement sequence followed by a spell cast command (MMSC)

To create the three level 3 actions, the parser listens for attack, move and spell cast commands. If a command is recorded, the current MM sequence is used as the sequence leading up to the command. As such, these three features are identical in the way they are recorded, but potentially record very different kinds of data. For example, the move command is sent much more often than the other two commands TODO.

2.2 Game-specific statistics

The other feature that was added for the machine learning models are game-specific statistics, which generally indicate the performance of the player. The statistics are:

- Gold per minute
- XP per minute
- CS per minute
- Denies
- Actions per minute
- Number of move commands on target
- Number of move commands on position
- Number of attack commands on target
- Number of attack commands on position
- Number of spell cast commands on target
- Number of spell cast commands on position
- Number of spell cast commands with no target
- Number of hold position commands

Some of these statistics are taken as per minute because the numbers can vary greatly depending on the length of the game.

3 Predicting the player

The first machine learning problem was one of binary classification. The model was trained on the question “Given a game of Dota 2, is the player on a specific hero the player we are looking for?” The problem was fixed on a particular hero in order to remove any extra complexity that may rise from including different heroes, such as playstyle of the hero.

3.1 Methodology

This problem was investigated with two different uses of the data. Initially, each individual level 3 action was taken as a single data point for training, rather than all level 3 actions in a single game. This meant there was no concept of a game, as the model simply took lists of level 3 actions instead. The three level 3 actions were each used individually to see which would be more indicative for

identifying the player. Each action was assigned as a positive sample if came from the particular player, and a negative sample otherwise.

In the second iteration of this problem, the level 3 actions were combined together so that a model could be trained on individual games as data points rather than actions. In this approach, a separate model was used for each of the level 3 actions as before, but the results of the three models are combined by a simple voting mechanism and a simple MLP classifier. This gave a better representation of identifying a player from a game from training, rather than a large list of level 3 actions.

3.2 Results

TODO graphs

4 Predicting the same player

5 Conclusion and future work

A lot of the foundational work has been completed, and a working model and machine learning pipeline has been established with good results. The goal of the next few weeks is to discover how to improve on the current results and find interesting relationships between the data and the predictive power of the models.

5.1 Heroes as a priori or feature

Currently, all machine learning has been done using the a fixed hero. This removes the addition factor of hero selection affecting the results and features, as different heroes have different playstyles. The next major piece of work is to incorporate hero selection as part of the pipeline, rather than excluding it.

5.2 Feature selection