CS486 Optional Course Project Proposal

Jason Sun

May 21, 2015

1 Introduction

This a project proposal to explore applications of machine learning into analysing the large quantity of data generated from each Dota game. I think this is an appropriate project for machine learning because it hass a large quantity of data available, and the dataa is changing with every release. And there are 15M unique players a month, with international contests for these games. So there is an interest in those games, like myself, to improve their chances of winning matches.

2 Game background

There are 102 heroes, and each hero has 101 matchup statistics against other heroes. There are two opposing teams, each of 5 heroes. And there are few game modes that are available, each can be potentially analysed. Complete descriptions on http://dota2.gamepedia.com/Game_modes.

- All Pick
- Single Draft
- Random Draft
- All Random
- Captains Mode
- Captains Draft
- Least Played
- Limited Heroes
- Ability Draft
- 1v1 Solo Mid
- All Random Deathmatch

There are many modes, some of them more used professionally, some of them less serious (for example, all random). In professional gaming world, quite frequently seen is Captain's Mode, a captain for the team has to to pick and ban heroes, taking turns and in sequence.

I'll focus on one mode for now in the sake of brievity. But it remains true that in general that the concepts we'll explore in this project remains applicable to the other modes. Concepts such as "what heroes should I pick for my team, knowing the other team has picked those heroes?"

3 My previous project

I have previously tried to make use of the public API that Valve (company that built the game) and do analysis. The source code repository is here github.com/sunapi386/dotabuff-ripper. In this codebase,

1. I attempt to scrape win rate data (which is summarized by a website called Dotabuff using an unknown and persumably proprietary algorithm). Usual win rate of hero A against hero B is in the range of $50 \pm 10\%$.

2. Loaded the scraped hero win rate data into a graph database and modeled each hero as a node and each edge as a win rate. Although probably not necessary, I was sold on the idea that a graph can model this data well and it would be fast to develop since I didn't need to write graph algorithms.

3. Made a simple command-line interface that queried the database: given that you have a set of heroes chosen, return a sorted set of heroes that your team plays the weakest against.. This was calculated by a very basic algorithm, which was: for each enemy hero chosen, I added their win rate against all the other heroes and returned a sorted list by decreasing winrate.

4. Finally, host this on the web so others can use it and potentially levy some donations for it. This last step was never done because I didn't think my algorithm was very sophisticated, and probably didn't perform much better than a player having choosing based on their experienced guess.

4 Proposed project

Instead of scraping the win rate data from Dotabuff, I wanted to scrape actual game matching data results and calculate my own win rate. My previous project was done in 2013 and I had a very basic understanding of data analaysis and algorithms. I think after two more years I can do predictions better. Some of the ideas I thought about exploring into and applying to analyse game match data were:

- Hidden Markov Models. As far as I know, this improved speech recognition industry by quite a bit because it was able to guess a new word based on the learned likelyhood of that word following a word just recognized. So in that sense we can train a HMM on the win rate dataset. I'm not sure how this is different from using neural networks.

- Neural Networks. This form is probably most appropriate, it is quite popular in machine learning, and has good results in pattern recognition. For example, we can create a training set based on the heroes that one team picked against another team and train which team was the winner, thus learning what are good team compositions against another team.

In terms of options A/B/C/D for the project, I think my proposal is mostly option B and some of C, which is to implement and experiment several AI techinques to solve this problem, and potentially develop a new AI techinque. Then evaluate theoretically and/or empirically the performance by taking it through actual matches. I think I'll start with option B as I'm not aware of a lot of potential ways to solve this problem and would like to try some existing techinques before calling it a day and creating one from scratch (which is presumably harder).