CS181 Practical 4: Swingy Monkey!

Victor Domene, Henrique Vaz, Joshua Meier

April 28th, 2016

1 Abstract

In this practical, we performed machine learning on Swingy Monkey, a game based on the viral game Flappy Bird that took off a few years ago. In Swingy Monkey, at any point in time, there are only two possible actions: to jump, or not to jump.

In this practical, we performed machine learning on truncated lists of user / artist play counts, in order to predict how many times each of these users listened to every artist in the collection. The implications of high accuracy here are quite interesting - there are many commercial applications to being able to predict a user's entire play history from just a few sample points. However, this also makes the problem extremely difficult. Unlike previous practicals where we had complete training data to work with, we did not necessarily have complete data for each user in the training set, and furthermore, we did not have a completeness metric for the data of any user. After performing a thorough data analysis to better understand the complexity of the data, we then continued to apply a variety of very different machine learning techniques, since many different intuitions could be applied here. We also performed feature engineering, but we were unable to incorporate these features (known as "side information" in the literature) into the algorithms in a fashion that allowed the algorithms to terminate in reasonable time. Ultimately, we discovered that a simple matrix factorization using just a single feature performed best, giving us the fourth position in the Public Leaderboard (as of 2pm on Friday). Notably, our hand-crafted matrix factorization algorithm outperformed several implementations from popular software packages.

2 Technical Approach

2.1 Dependencies

2.2 Data Analysis and Centering

2.3 Feature Engineering: Handling the State Space

2.3.1 Choosing the Features

2.3.2 Binning

2.4 Q-Learning

2.4.1 ϵ-greedy

2.5 Parameter Tuning

2.5.1 Learning Rate: α

2.5.2 Discount Rate: γ

2.5.3 Randomness: ϵ

2.5.4 Number of Bins

2.6 Supervised Approach: For Fun!

3 Code

Our code for this practical can be found in this GitHub Repository. There is a README file with some setup instructions.