Predicting the Outcome of College Basketball Tournament Using a Least-Squares Network Model

Yosuke Sugishita
CS490
yosuke.sugi@gmail.com

Henry Li
CS490
henry.s.leigh@gmail.com

Abstract

Ranking sports teams and predicting their games' outcome is often said to be a difficult problem. In this paper, we propose a simple least squares network model to rank teams in the American college basketball league, and attempt to predict the final tournament's outcome. By training on the past 18 tournaments, we find that our model accurately predicts the outcome of a game in the final tournament with $70(\pm 4)\%$ accuracy. We first train the model using a basic least square model, and proceed to train the same model using ℓ_2-regularization, which stabilizes our predicted outcome of a new game.

1 Introduction

1.1 Background Information (Description of Data)

The biggest tournament in the American college basketball league is NCAA Basketball, which is commonly referred to as "March Madness". It is a single-elimination tournament played by 67 teams, meaning there are 66 games' outcome to predict. Roughly 3000 games are played amongst more than 300 teams in our data set every year during the regular season, which we are going to use for tranining our model.

1.2 Objective (Predictive Accuracy)

Our objective is to predict this year's (the 2014 season) tournament's outcome and we aim to maximize the simple predictive accuracy of win/lose. Therefore, we are going to use past years' tournament results as our test data set. The tested accuracy of a model is defined as follows:

$$\text{tested accuracy} = \frac{\text{number of correct predictions}}{\text{number of all predictions}} \tag{1}$$

2 Model

2.1 Model Construction

We define differences

$$\hat{d}_{ij} = \hat{x}_i - \hat{x}_j$$

where $\hat{x}_k$ is the 'potential' for a team k. These differences represent the difference in final score between a pair of teams i, j given that they have played a game against each other. Now if $\hat{d}_{ij} > 0$, we know that team x_i had won, since $\hat{x}_i > \hat{x}_j$. Using this encoding, we can formalize an equation with which we can form predictions. With $\hat{d}$ as an estimator of the resulting score, we can minimize the square differences

$$\min_{x_i} \sum_{\texttt{all games}} \left(d_{ij} - \hat{d}_{ij}\right)^2$$

2.1.1 Training the Model

2.1.2 Testing the Prediction Accuracy

2.2 Confidence Interval Analysis

2.3 Use of ℓ_2-Regularization

3 Results

3.1 Predictive Accuracy

We achieved roughly $70(\pm 4)\%$ accuracy for predicting the past 18 tournaments' outcome (for win/lose prediction). This is an out-of-sample result as we trained our model on the same seaosn's regular game results, which preceed the tournament.

3.2 Regularization (Irregular Behaviour in the Results)

We noticed that differences between our potential estimates are sometimes unusually high or low (for instance, in the order of 10^9). One can see this effect easily from the average standard deviation of potential estimates for each year.

`<TODO TABLE: show the standard deviation>`

As the potential differences are used as the predicted score difference of a game, this is clearly an irregular result. Also, as we examined the plot of predicted score differences against actual scores, we realized that the predictions can be unstable particularly when the actual score differences are small (when the games are "close").

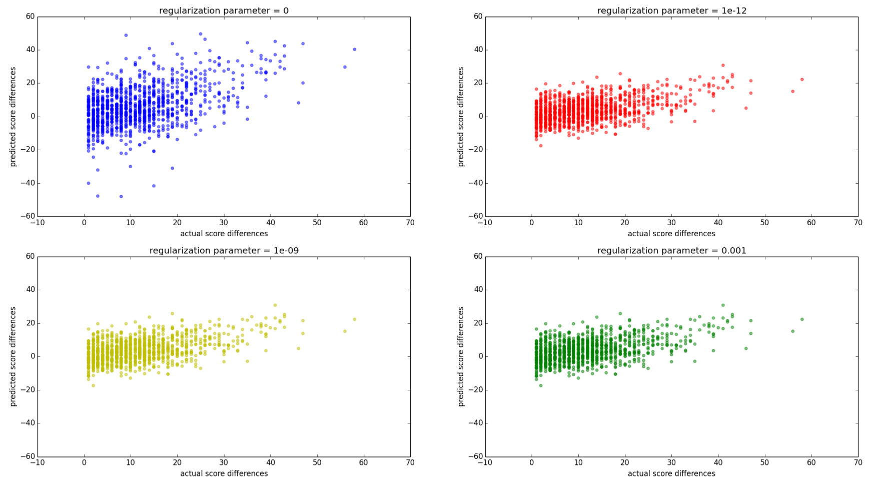

Figure 1: Predicted score differences against actual score differences

To prevent these unusual behaviors and also to stabilize the predictions, we utilized ℓ_2-regularization. The model is described in Section 2.3: ℓ_2-Regularization. As expected, this model yielded more stable predictions as shown in Figure 1. This was also clear when we compared the correlation coefficients between actual score differences and predicted score differences (it improved from 0.456 to 0.468. However, it did not change the predictive accuracy significantly.

Interestingly, ranging the regularization parameter from 10^{-9} to 0.001 hardly changed the results as can be seen from Figure 1.

4 Conclusion

4.1 Summary

4.2 Further Considerations