CSE 446 Final Project Milestone

Damir M. Zhaksilikov
Department of Computer Science
University of Washington
Seattle, WA 98105
damir@cs.washington.edu

Jason Frazier
Department of Computer Science
University of Washington
Seattle, WA 98105
jasonf56@cs.washington.edu

Abstract

The aim of this experiment is to develop a model based on Logistic Regression to successfully predict the outcome of the 2017 NCAA March Madness tournament. As a part of Kaggle's March Machine Learning Mania 2017 challenge, we were provided with initial data for regular and tournament game scores as well as tournament seedings and slots for the 2003 to 2016 seasons [1]. Our course of action is to first confidently predict regular season games using logistic regression coupled with feature selection techniques. Next, we plan to experiment with additional datasets to move towards predicting tournament outcomes using regular season data. These datasets would supplement our isolated features to gear our model towards predicting tournament results rather than regular season games. After these adjustments, our model will be tested against the 2014-16 March Madness Tournament results.

1 Current implementation

Current experiments have been focused on formatting and normalizing regular season data. Our team developed a L2 Regularized Logistic Regression model and ran experiments on the processed data.

1.1 Data: season averages

Scripts to format and normalize regular season data were written using Python's SQLite library. The goal of this processing was to replace each games actual performance statistics with team average statistics. These averages, in turn, were used predict each game's outcome.

The original regular season data contains a row for each game and includes winning and losing team's statistics per game. Our script averages the statistics for each team during a given year. As each row denotes teams by a win or loss, the script sums the statistics for each team's winning and losing columns separately. Each team's winning and losing data is then combined and averaged over the number of the total games.

For each game, each team's actual statistics are replaced with their averaged statistics. A result value was added to denote the outcome of the game:

1 if team1 wins
-1 if team2 wins

Two flipped rows in the processed data exist for each row in the original data. In one row, the winning and losing teams are denoted as team1 and team2 respectively (result has a value of 1). The other denotes the losing team as team1 and winning team as team2 (result has a value of -1). Intuitively, team1 and team2's statistics should have an inverse relationship, therefore flipping teams would result in an inverse classification.

1.2 Logistic regression: predicting regular season games

Our learner uses a L2 Regularization Logistic Regression model. This model is implemented using stochastic gradient descent. Data from the 2016 season was processed using the methods described above and then used to experiment with varying numbers of steps and regularizing coefficients.

1.2.1 Measure of quality

We measured the quality of our model's prediction by calculating the percent of games correctly classified. The regular season data was shuffled and 4/5 of the data was used for training and other 1/5 was used for testing. The percent correctly classified in the test set was our result, it is as follows:

$$\frac{1}{n}\sum_{i=1}^{n} 1[y_i = \hat{y}]$$

Equation 1: Percent Correctly Classified (PCC) Equation

1.2.2 Experiment 1: lambda value

To determine the best regularization coefficient, the weights were calculated with one iteration through the 2016 data with step size of 0.0001. The PCC for each coefficient was averaged over 3 trials. A regularization coefficient of 16 yielded the best PCC.

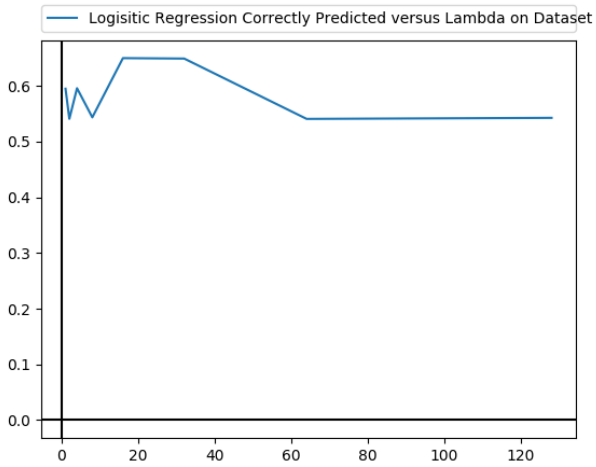

Figure 1: Graph of PCC versus Lambda

Lambda	PCC
1	0.59509392
2	0.54106505
4	0.59602546
8	0.54370439
16	0.65005434
32	0.64927806
64	0.54090980
128	0.54277286

Figure 2: Experiment 1 Results

1.2.3 Experiment 2: number of steps vs. PCC

Using a fixed regularization coefficient of 16, we experimented with how our PCC changed as we increased the number of iterations over the training dataset made by our stochastic gradient descent:

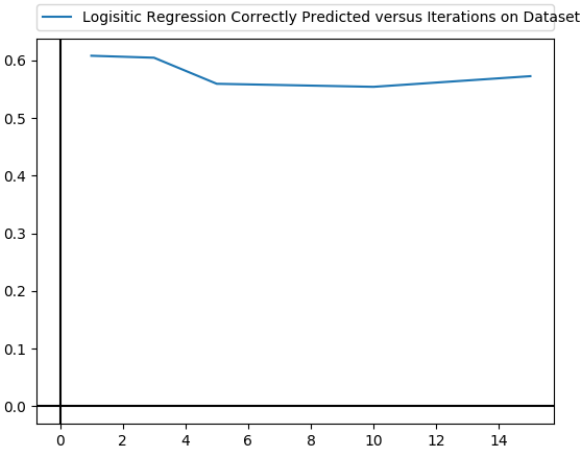

Figure 3: Graph of Iterations versus Lambda

Lambda	PCC
1	0.60782487
3	0.60440925
5	0.55922993
10	0.55395125
15	0.57242664

Figure 4: Experiment 2 Results

1.2.4 Analysis of results

At this midpoint, our model predicts about 60% of regular season games correctly. While this is a decent start, it is important to keep in mind that this provides little insight on how this will translate to tournament games. The features that had the largest weights were the location of the game and average score difference per team. These factors will be important to note as we begin feature selection.

One unexpected result arose from our experiments, the resulting weights did not have an inverse relationship between team1 and team2. This leads us to believe that either our data was formatting incorrectly or that our model will need to be adjusted. This is a lead that we will need to explore immediately.

2 Future implementation

2.1 Expand experiments

Current experiments have limited to the 2016 regular season due to time constraints. Thus, the learned model is skewed. Expanding experiments to all regular season data is a necessary next step before our team considers feature selection and/or supplemental data.

2.2 Feature Selection

Some statistics in the provided regular season data may be redundant and/or non-impactful. We plan to experiment with the Scikit-Learn Python library to reduce the number of team statistics without increasing our error.

2.3 Supplement dataset

Although regular season data is valuable in predicting a game's outcome, supplemental data is necessary to address the additional variables of tournament games. We will experiment with various potential data supplements and choose the most successful in minimizing our SSE. These will not be used to predict regular season games as they are not available during regular season games.

2.3.1 Tournament seeding

The Kaggle dataset provides tournament seeding. Tournament seeding is done by a selection committee whose expertise in team analysis is assumedly precise. The viability of seeding in prediction models was explored during the 1996 NCAA March Madness by Schwertman, Schenk, and Holbrook's [2]. The tournament seeding feature will simply be a number value denoting the seed of the team. We will need to update our SQL script to include this data.

2.3.2 Las Vegas spread

The Las Vegas spread provides a majority consensus (for a large population) on the prediction for the result of a game. As explained by Lopez and Matthews, "rules of efficient gambling markets imply that, over the long run, it is nearly impossible to outperform the point spreads set by sportsbooks in Las Vegas" [3]. Due to a similar hypothesis, we believe that due to the large population consensus, the Las Vegas spread will be a beneficial feature for our model. Notably, this feature will measure the magnitude of the spread between teams, rather than denoting a predicted win by team1 or team2. These will be manually entered.

2.4 Bracket builder

The goal of our model is to predict the most probable bracket for this year's March Madness tournament. Due to the nature of tournaments, we will not have all matchups available to us. Instead, we plan to consider all possible brackets and determine their probability. The bracket with the highest probability will be selected. For all possible brackets, we will create a row where each winner (as noted in the possible bracket) is team1 and the other is team2. Thus, the probability of the bracket is the probability of each team1 winning each respective game

$$\prod_{i \in PotentialGames} sigmoid(w^T x_i)$$

Equation 2: Bracket Probability

References

[1] March Machine Learning Mania 2017 | Kaggle. (n.d.). Retrieved February 20, 2017, from https://www.kaggle.com/c/march-machine-learning-mania-2017

[2] Schwertman, N. C., Schenk, K. L., & Holbrook, B. C. (1996). More Probability Models for the NCAA Regional Basketball Tournaments. The American Statistician, 50(1), 34-38. Retrieved February 20, 2017.

[3] Lopez, M. J., & Matthews, G. J. (2015). Building an NCAA men's basketball predictive model and quantifying its success. Journal of Quantitative Analysis in Sports. Retrieved February 20, 2017