March Madness Prediction: A Matrix Completion Approach

Hao Ji, Erich O'Saben, Adam Boudion, and Yaohang Li
Department of Computer Science
Old Dominion University
hji@cs.odu.edu, eosab001@odu.edu, aboudion@cs.odu.edu, yaohang@cs.odu.edu

ABSTRACT

In this paper, we present a new predictive model based on matrix completion to forecast the winning probabilities of each potential game in NCAA basketball tournament. The predictive model is evaluated in Kaggle's March Machine Learning Mania competition, where each submission is rated by Log Loss function for each matchup that occurs in the tournament. We discuss neural network and probability refinements used to estimate and normalize our predicted probability from the predicted performance accomplishments in each matchup and team ranks, with the ultimate goal of lowering our Log Loss score per matchup. In the stage one of predicting 2011-2014 results, our team is ranked 61 out of 347 teams with Log loss score of 0.56915, which exceeds Seed-based benchmark method (0.59071). Furthermore, we analyze the pitfalls that were encountered during our research so others can improve upon our methodology for future research into the March Madness tournament.

Keywords: Matrix Completion, Neural Networks, March Madness Prediction, Singular Value Thresholding, March Machine Learning Mania.

1. INTRODUCTION

The NCAA Men's Division I Basketball Tournament, or commonly referred to as March Madness, is one of the most popular annual sporting events in the United States. Each year, 68 teams are selected for a single elimination, playoff style tournament with the final two teams competing in the championship game. Millions of people have submitted brackets to March Madness tournament pools to compete for winning prediction prizes. The March Machine Learning Mania competition hosted by Kaggle.com is one of free and legal tournament pools, where the bracket prediction requires us to submit probabilities of every possible matchup of tournament. The Log Loss function (the predictive binomial deviance) is used to judge each submission,

$$LogLoss = -\frac{1}{n}\sum_{i=1}^{n}(y_i \log(p_i) + (1 - y_i)\log(1 - p_i))$$

where n is the number of games, p_i is the winning probability of team 1 playing against team 2, and y_i equals 1 if team 1 wins over team 2 and 0 otherwise. A smaller value of $LogLoss$ indicates better performance of the predictive model.

In order to fill out tournament brackets with high predictive accuracy, many computer simulations and algorithms have been developed to model the tournament and attempt to explore the effective strategies for March Madness prediction. For instances, the Colley method [9] and the Massey method [10] are two early work using statistical methods to predict the outcome of tournaments. Later on, Smith and Schwertman [7] proposed a regression model and found the nearly-linear relationship between teams' seeds and tournament results. More recently, Gupta[12] used a dual-proportion probability model with rating of teams learned from season games. Lopez and Matthews[13] designed a logistic regression model by taking advantage of team-based possession metrics, which won the 2014 Kaggle competition. Ruiz and Perez-Cruz proposed modified a classical model for forecasting soccer to predict basketball game and stated that high predictive performance obtained [11]. We refer to [12, 14] for a rich overview of existing literature.

In this paper, we design a new predictive model using matrix completion to predict the results of the NCAA tournament. First of all, we formulate performance details from regular season games of the same year into matrix form, and apply matrix completion to forecast the potential performance accomplishments by teams in tournament games. Second, we project the predicted performance accomplishments into matchup scores, where the relationship between performance accomplishments and scores are modeled via neural network using historical seasons and tournament data. Third, probability adjustments are carried out to derive the appropriate winning probabilities from the estimated matchup scores.

The rest of the paper is organized as follows. In Section 2, March Madness prediction is formulated as Matrix Completion problem. Section 3 describes the proposed predictive model. The results of our submission reported in section 4. Finally, Section 5 summarizes the paper.

2. PREDICTING AS MATRIX COMPLETING

Incomplete matrices with the presence of missing entries often arise in the situations where data are unknown or unobservable. For instance, in the Netflix problem, as most users rate only a small subset of movies, rating matrices appear to be very sparse and contain a large amount of unknown ratings [4,5]. The objective of matrix completion is to recover the missing (unknown) entries of an incomplete matrix from a small subset of observed ones [1-3]. It is commonly believed that the most actions of matrices are effected by only a few factors in real-life applications. Therefore, an important but natural assumption is set with the matrix completion problem that the matrix to recover is of low rank or nearly low rank. Let M denote an incomplete matrix and Ω be a set of indices of observed entries, the matrix completion problem is then defined as finding a low-rank solution X to the following optimization problem,

$$\min_{X} \|X\|_{*}$$
$$\text{subject to } \mathcal{P}_{\Omega}(X) = \mathcal{P}_{\Omega}(M)$$

where $\|\cdot\|_{*}$ is the nuclear norm which is the sum of singular values and $\mathcal{P}_{\Omega}$ is the projection operation defined as

$$\mathcal{P}_{\Omega}(X)_{ij} = \begin{cases} M_{ij} & if\ (i,j) \in \Omega \\ 0 & if\ (i,j) \notin \Omega. \end{cases}$$

Many numerical algorithms have been developed in the literature to solve the above matrix completion problem. For example, convex optimization algorithms based on Semi-definite Programming to fill out the missing matrix [1,2] and the Singular Value Thresholding (SVT) algorithm to efficiently approximate the optimal result[3]. We refer to [2] for more comprehensive overview on nuclear norm minimizations.

In the same spirit, March Madness prediction can be formulated as matrix completion problem as well. Figure 1 shows the plot of a matrix of games played between 364 different college basketball teams in regular season 2015, where teams are placed on rows and columns, and a blue dot indicates the game has played between team i and team j.

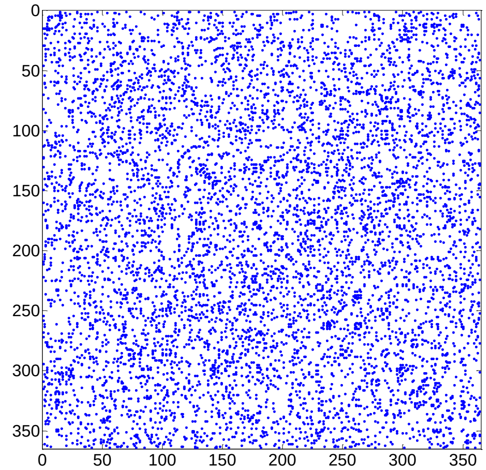


Figure 1. Games Played Between 364 Different College Basketball Teams in Regular Season 2015

As only 3771 matches were held in regular season 2015, one can find that the matrix of games is sparse with most of entries are unknown. If those missing entries can be recovered, the outcome of each potential matchup in the tournament can be estimated by assigning with the corresponding results from the completed matrices.

It is well known that the outcome of a basketball game depends to a large extent on the following performances accomplishments made by teams,

1) field goals attempted (fga);
2) field goals made (fgm);
3) three pointers attempted (fga3);
4) three pointers made (fgm3);
5) free throws attempted (fta);
6) free throws made (ftm);
7) offensive rebounds (or);

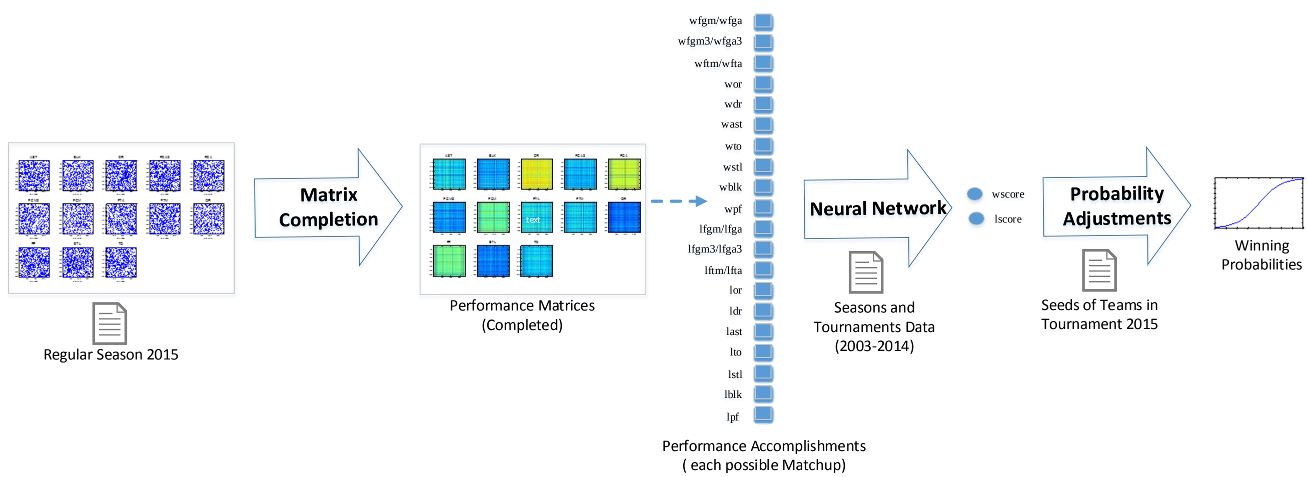


Figure 2. Procedure of the Proposed Predictive Model

8) defensive rebounds (dr);
9) assists (ast);
10) turnovers (to);
11) steals (stl);
12) blocks (blk);
13) personal fouls (pf).

Under the assumption that the strengths and weaknesses of any team can be reflected from game records in regular season prior to the tournament, we place the related performance data from regular season of the same year into 13 performance matrices to predict outcome of a basketball game. The matrix completion is then applied to complete each performance matrix in our predictive model.

Figure 2 shows the procedure of the proposed predictive model, where the upper case of performance name is used to denote each performance matrix, for example, FGA represents a matrix of field goals attempted. The predictive model proposed in this paper consists of three phases: (1) matrix completion, which predicts the performance accomplishments in every possible tournament game. (2) neural network, where the predicted performance accomplishments are used to estimate matchup scores based on the relationship learned from historical records. (3) probability adjustments, where the predicted winning probability are derived from the estimated matchup scores and team ranks.

3. METHODS

3.1 Matrix Completion

We apply Cai's SVT algorithm [3] to complete each performance matrix. Taking the matrix FGA of field goals attempted by teams in regular season 2014 as example, where Ω is assigned with a set of indices of the played games. The SVT algorithm seeks a low-rank matrix X to minimize the following Lagrange dual function,

$$\tau\|X\|_* + \frac{1}{2}\|\mathcal{P}_\Omega(X) - \mathcal{P}_\Omega(FGA)\|_F^2$$

where $\mathcal{P}_\Omega$ is the projection operation and τ is a Lagrange multiplier trading off between the nuclear and Frobenius norm. In general, suppose the matrix to recover is of size $m \times n$, the value of τ is set to be a factor of $\sqrt{mn}$, such that $\tau = \omega\sqrt{mn}$, where ω is a positive number.

A difficulty in applying the SVT algorithm to March Madness prediction is that not all values of ω can make the SVT algorithm provide a satisfactory completed matrix. Figure 3 shows the predicted field goals attempted by the SVT algorithm at $\omega = 8$, $\omega = 120$, and $\omega = 150$, respectively. We plot the real ones of tournament 2014 in red color for comparison purposes. One can find that at very small value ω such as ω=8, the completed matrix is polluted with small, even negative values, which do not make much sense in basketball games. In order to obtain a reliable completed matrix for all performance matrices, we use the average of 50 completed FGA matrices from using the SVT algorithm at different ω from 101 to 150. (Figure 4).

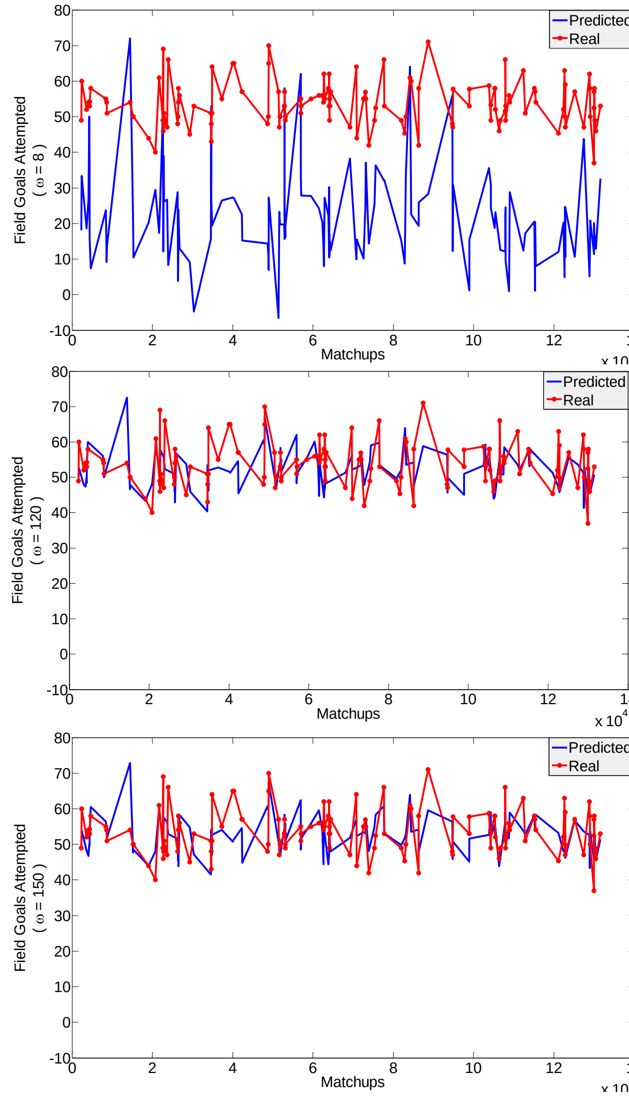


Figure 3. The Predicted and the Real Field Goals Attempted by SVT at ω=8 (Upper), ω=120 (Middle) , and ω=150 (Lower)

In the same way, we complete other 12 performance matrices based on records in regular season of the same year. After matrix completion, the corresponding entries for one tournament matchup are selected from each performance matrix to form a vector of performance accomplishments, which will be input into a trained neural network to generate the estimated matchup scores, as shown in Figure 2.

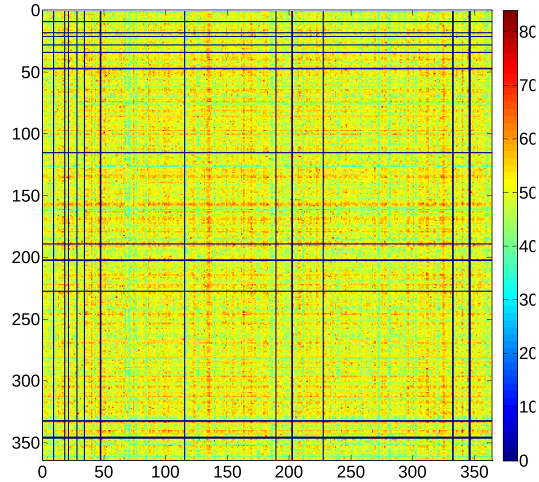


Figure 4. The Completed Matrix FGA

3.2 Neural Network

We use a feed-forward neural network to model the relationship between performance accomplishments and scores. Figure 5 illustrates architecture of a neural network used with 15 neurons.

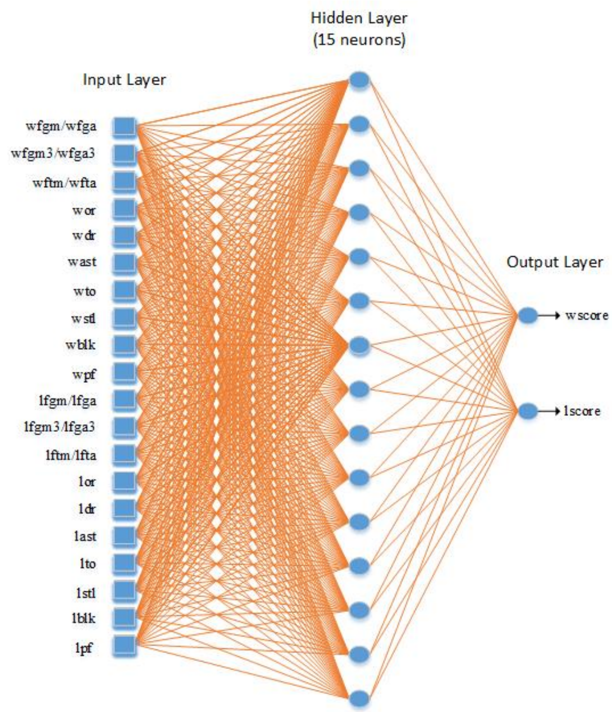


Figure 5. The Architecture of Feed-Forward Neural Network

For training neural network, the dataset is selected from team-level historical season and tournament records from 2003 to 2014. We randomly divide the dataset into three subsets, 70% samples are used for training, 15% data for validation, and 15% data for testing. As shown in Figure 5, twenty performance accomplishments by the winning team and the losing team, including field goals percentage (fgm/fga), three pointers percentage (fgm3/fga3), free throws percentage (ftm/fta), offensive rebounds (or), defensive rebounds (dr), assists (ast), turnovers (to), steals (stl), blocks (blk), and personal fouls (pf), are encoded as the network input, while the corresponding network target is set to two scores. Prefixes "w" and "l" are used to distinct between the winning team and the losing team. We also randomly flip the order of the winning team and the losing team in network input and target to increase the learning capability of neural network.

Once the neural network is trained, the predicted scores of each possible matchup in tournament can be estimated with ease based on the forecasted performance accomplishments from the previous matrix completion step.

3.3 Probability Adjustments

Although normalized scores can be used as probabilities, such that

$$p_{team1,team2} = \frac{score_{team1}}{score_{team1} + score_{team2}} \qquad (1)$$

where $score_{team1}$ and $score_{team2}$ denote the points scored by team 1and team 2, respectively, it cannot accurately reflect real differences among matchup scores. For example, suppose that team 1 wins over team 2 with points 80 to 50,

$$p_{team1,team2} = \frac{80}{80 + 50} = 0.6154,$$

the computed winning probability 0.6154 from equation (1) is too low. As we known, beating by 30 more points in a basketball game means a dominating advantage. Therefore, in order to generate a more reasonable probabilities, we use equation (2) with a sixth power of a score instead,

$$p_i = \frac{score_{team1}^6}{score_{team1}^6 + score_{team2}^6} \qquad (2)$$

which gives

$$p_{team1,team2} = \frac{80^6}{80^6 + 50^6} = 0.9437,$$

so that the significant difference between scores can be retained. In addition, to increase the predict accuracy, we use equation (3) to refine probabilities in our predictive model, which takes teams' seeds into account,

$$p_i = \frac{1}{2}\left(\frac{1}{2} + \frac{3(seed_{team1} - seed_{team2})}{100} + p_i\right) \qquad (3)$$

where $seed_{team1}$ and $seed_{team2}$ denote the rank of team 1and team 2 in the tournament, respectively.

While Kaggle allows participants submit up to two brackets and the final leaderboard score will be chosen based on the best one. To attempt to further lower the *LogLoss* value of our predictive model, we impose the following two equations (4) and (5) on deriving the aggressive probabilities as our second bracket,

$$p_i = \begin{cases} 1 & if\ \ Seed_{team1} = 1 \ \ and \ \ Seed_{team2} = 16 \\ 0 & if\ \ Seed_{team1} = 16 \ \ and \ \ Seed_{team2} = 1 \end{cases} \quad (4)$$

$$p_i = \begin{cases} 0.9545 & if\ \ score_{team1} - score_{team2} > 20 \\ 0.0455 & if\ \ score_{team1} - score_{team2} < -20 \end{cases} \quad (5)$$

where equation (4) is constructed from tournament statistics [8], no team with seed 16 has ever won a team with seed 1, while equation (5) is used if a team gains more than 20-points advantage in our model.

4. RESULTS

The March Machine Learning Mania competition consists of two stages: In the first stage, the participants develop and test their models on predicting results of tournaments from 2011 to 2014. In the second one, the participants predict the outcome of 2015 tournament.

Our initial bracket based on (2) for predicting tournaments in 2011-2014 has *LogLoss* value of 0.61. By applying (3), the LogLoss value decreases down to 0.57570. A further reduction is gained by using our aggressive bracket, which lets our team be ranked 61 out of 347 teams with Log loss score of 0.56915, which

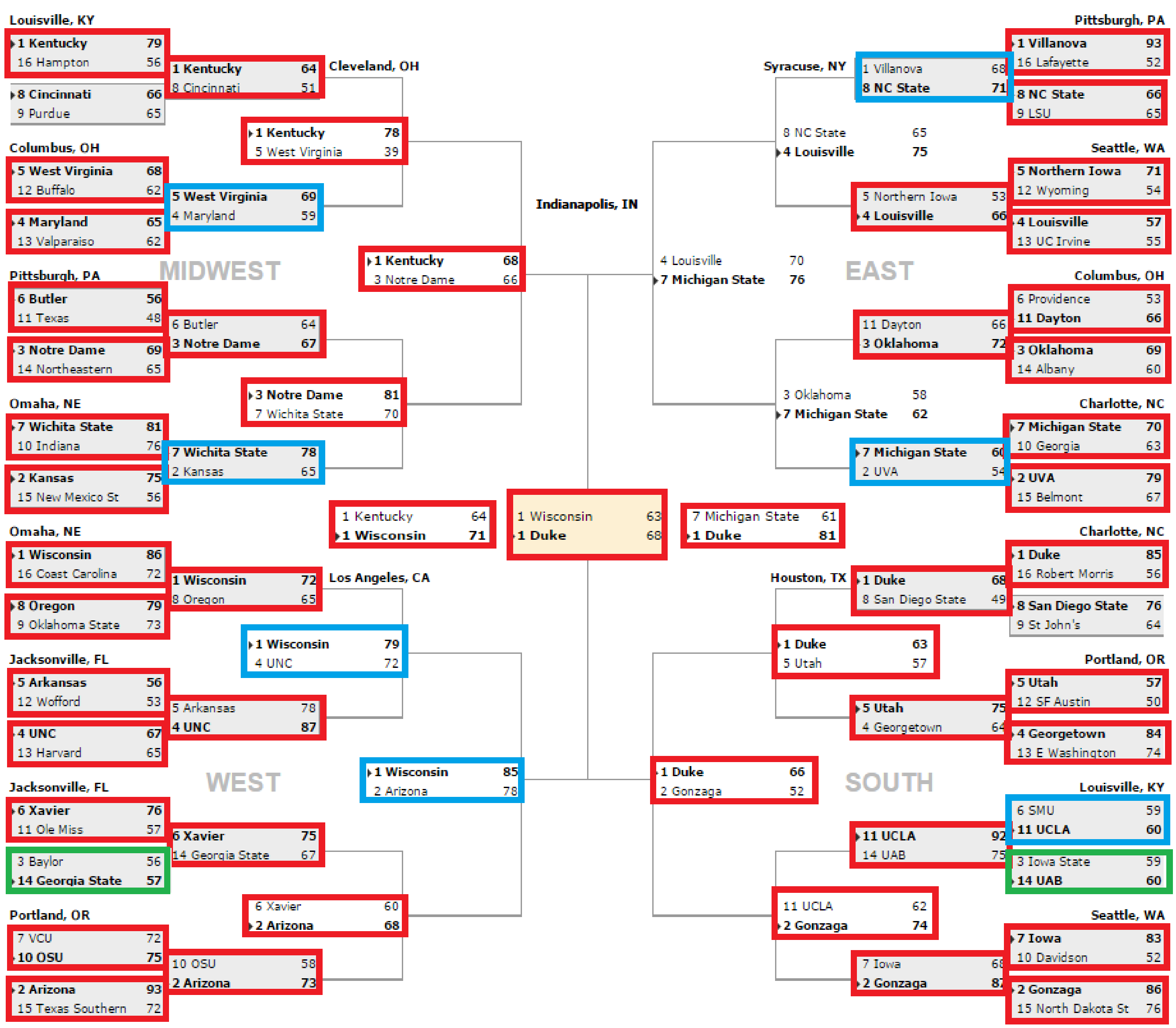


Figure 6. The Actual Result of March Madness 2015

exceeds Seed-based benchmark method (0.59071).

For the second stage, we submitted the winning percentages of 2278 potential games of tournament 2015. For simplicity, the standard bracket and the aggressive one of forecasting the tournament outcome are shown in Appendix, where the results are generated according to our predicted winning probabilities.

Along with the championship game of March Madness 2015 completed on April 6th, 2015, our brackets finish with the $LogLoss$ score of 0.529547, where $LogLoss$ is evaluated based on the probabilities on the actual tournament games. On the happy side, we successfully predict the win/lose results on 49 out of 63 tournament games, and we particularly predict correctly in the game between Wisconsin and Kentucky. Figure 6 shows the actual result of the March Madness 2015 [15], where the games we predicted correctly are highlighted in red color. However, our submitted brackets are heavily penalized by several upsets, as shown in green color and blue color in Figure 6. For example, in the games (green color) that No. 14 seed UAB drops No. 3 seed Iowa State and No. 14 seed Georgia State defeats No. 3 seed Baylor, where we bet the winning probabilities 0.95 and 0.8 on Iowa State and Baylor, respectively, which result in two largest increments in our $LogLoss$ score of 0.047 and 0.0312.

5. CONCLUSIONS

In this paper, we present a matrix completion approach to predict the performance accomplishments of every possible matchup in March Madness competition. An elaborated neural network and probability adjustments are carried out to estimate the winning probability of each game based on the predicted performance details and team ranks.

There is a lot of space to improve our predictive method. For example, in our current model, incomplete performance matrices are completed individually without considering any potential correlations and team seeds, which significantly hampers the accuracy of our brackets. In our predictive model for next year, we plan to pre-process performance details based on teams ‘seeds and construct a single larger matrix to treat all those key information together. Moreover, a more careful probability adjustments will be designed to set up the winning probabilities.

ACKNOWLEDGEMENTS

Hao Ji acknowledges support from ODU Modeling and Simulation Fellowship.

REFERENCES

[1] Candès, Emmanuel J., and Benjamin Recht. "Exact matrix completion via convex optimization." Foundations of Computational mathematics 9.6 (2009): 717-772.

[2] Recht, Benjamin, Maryam Fazel, and Pablo A. Parrilo. "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization." SIAM review 52.3 (2010): 471-501.

[3] Cai, Jian-Feng, Emmanuel J. Candès, and Zuowei Shen. "A singular value thresholding algorithm for matrix completion." SIAM Journal on Optimization20.4 (2010): 1956-1982.

[4] Rajaraman, Anand, and Jeffrey David Ullman. Mining of massive datasets. Cambridge University Press, 2011.

[5] Bennett, James, and Stan Lanning. "The netflix prize." Proceedings of KDD cup and workshop. Vol. 2007. 2007.

[6] Kaggle’s March Machine Learning Mania competition for the 2015 NCAA Tournament. available online: http://www.kaggle.com/c/march-machine-learning-mania-2015

[7] Smith, Tyler, and Neil C. Schwertman. "Can the NCAA basketball tournament seeding be used to predict margin of victory?." The American Statistician 53.2 (1999): 94-98.

[8] NCAA Men's Division I Basketball Championship. available online: http://en.wikipedia.org/wiki/NCA-A_Men%27s_Division_I_Basketball_Championship

[9] W. N. Colley, Colley’s Bias Free College Football Ranking Method: The Colley matrix explained, (2002) available at: http://www.colleyrankings.com /method.html.

[10] K. Massey, Statistical models applied to the rating of sports teams, undergraduate honors thesis, Bluefield College, 1997.

[11] Ruiz, Francisco JR, and Fernando Perez-Cruz. "A generative model for predicting outcomes in college basketball." Journal of Quantitative Analysis in Sports 11.1 (2015): 39-52.

[12] Gupta, Ajay Andrew. "A new approach to bracket prediction in the NCAA Men’s Basketball Tournament based on a dual-proportion likelihood." Journal of Quantitative Analysis in Sports. 11.1 (2015): 53-67.

[13] Lopez, Michael J., and Gregory Matthews. "Building an NCAA mens basketball predictive model and quantifying its success." arXiv preprint arXiv: 1412.0248(2014).

[14] Glickman, Mark E., and Jeff Sonas. "Introduction to the NCAA men’s basketball prediction methods issue." Journal of Quantitative Analysis in Sports 11.1 (2015): 1-3.

[15] NCAA Tournament Bracket – 2015. available online: http://espn.go.com/mens-college-basketball/tournam-ent/bracket.

BIOGRAPHIES

Hao Ji is a Ph.D. student in the Department of Computer Science at Old Dominion University. He received the B.S. degree in Applied Mathematics and M.S. degree in Computer Science from Hefei University of Technology in 2007 and 2010, respectively. His research interest include High Performance Scientific Computing, Monte Carlo Methods, and Big Data Analysis.

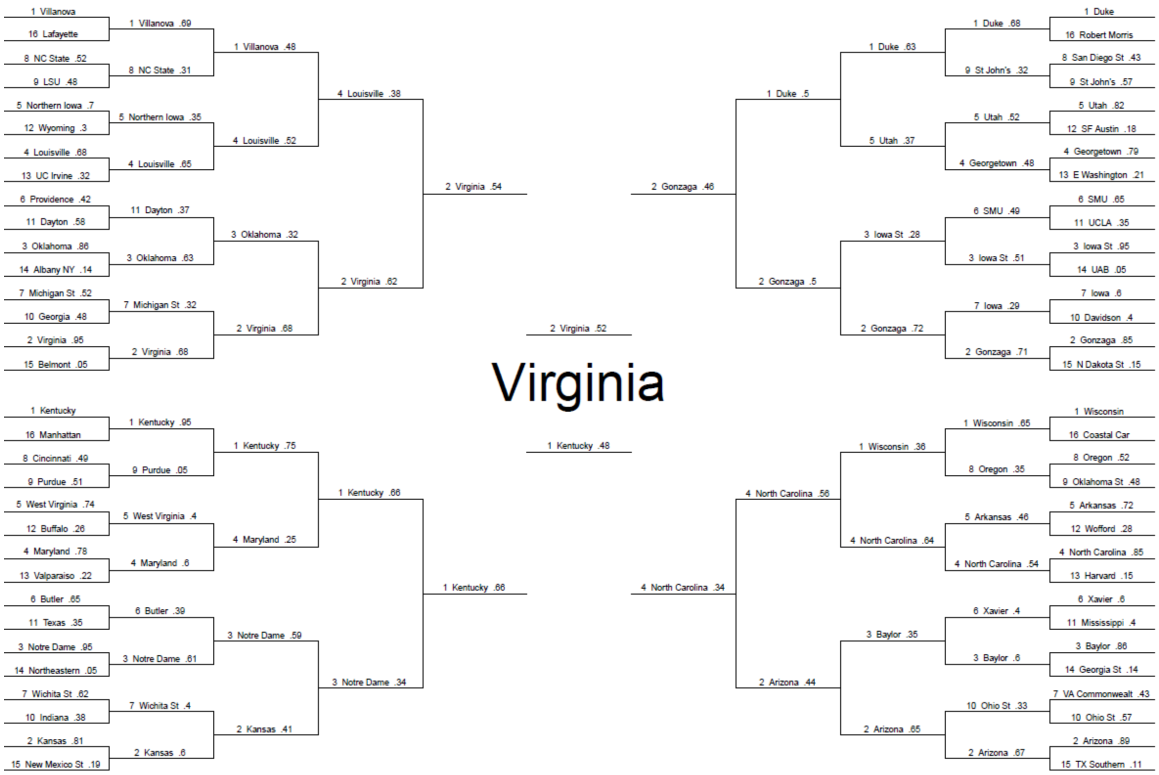
Erich O'Saben is an undergraduate Computer Science student at Old Dominion University. Erich is an aspiring data scientist with interests in artificial intelligence, machine learning, and predictive analytics.

Adam Boudion is an undergraduate student in the Department of Computer Science at Old Dominion University in the class of 2015. He also received a B.S. degree in Biology and Chemistry from Christopher Newport University in 2009. His research interests include applications of artificial intelligence.

Yaohang Li is an Associate Professor in Computer Science at Old Dominion University. He received his B.S. in Computer Science from South China University of Technology in 1997 and M.S. and Ph.D. degrees from the Department of Computer Science, Florida State University in 2000 and 2003, respectively. After graduation, he worked as a research associate in the Computer Science and Mathematics Division at Oak Ridge National Laboratory, TN. His research interest is in Computational Biology, Monte Carlo Methods, and High Performance Computing.

APPENDIX

(1) The Standard Bracket



(2) The Aggressive Bracket