LRMC Model and SVM for NCAA Basketball Prediction

Ying Li and Hoanh Nguyen

Introduction

Each year, more than $3 billion is wagered on the NCAA basketball tournament. Most of that money is wagered in pools where the object is to correctly predict winners of each games. The purpose of this project is to try to predict future games using historical data.

At the conclusion of each college basketball season, the NCAA holds a 64-team tournament. The participants are the champions of the 31 conferences in Division I, plus the best remaining teams (as judged by the tournament selection committee). In addition to choosing the teams, the selection committee also seeds them into four regions, each with seeds 1–16. The four teams judged best by the committee are given the No. 1 seeds in each region, the next four are given the No. 2 seeds, etc. Within each region, the 16 teams play a four-round single-elimination tournament with match-ups determined by seed (1 vs. 16, 2 vs. 15, etc.); the winner of each region goes to the Final Four. The Final Four teams play a two-round single-elimination tournament to decide the national championship. Throughout all six rounds of the tournament, each game is played at a neutral site rather than on the home court of one team or the other.

Figuring out which team is better than another is a difficult thing to do. We will never have every relevant bit of information such as player data or information on coaches. Another thing we don't have is a lot of data to work with. What we do have are the outcomes of previous games. We must use this data to either compute a way of ranking teams or comparing them against each other to see which would win on a neutral court during the tournament.

Background

Many have tried to predict the outcome of the NCAA basketball tournament. As mentioned earlier a lot of money is wagered on this tournament. A number of people go with gut feeling or devise a way of picking winners. Some use machine learning to learn from past data in order to predict the future.

The paper titled “A logistic regression/Markov chain model for NCAA basketball” outlines one way or calculating the likely winner of a game. The first thing that is done is to calculate the transition probabilities using a logistic regression model. This will estimate the probabilities that a team will win with a certain margin against another team at home. Because there are sometimes large outliers logistic regression is used to smooth out the probabilities. Those transition probabilities are then used to seed the Markov chain transition matrix which will provide the probabilities of one team being better than another.

Other machine learning techniques such as decision trees, rule learners, artificial neural networks, naïve bayes, and ensemble learners where evaluated in the paper titled “Predicting NCAAB match outcomes using ML techniques.” First they calculated a full set of raw statistics, normalized statistics, and adjusted efficiencies. Next they used some tools to help identify the most useful statistics for predicting the outcome of matches. Finally they applied the machine learning algorithms to the dataset.

Approach

Two approaches were taken, Logistic Regression/Markov Chain Model (LRMC) and Support Vector Machines SVM with simple statics.

We begin with a Markov chain with one state for each team. The intuition is that state transitions are like the behavior of a hypothetical voter in one of the two major polls. The current state of the voter corresponds to the team that the voter now believes to be the best. At each time step, the voter reevaluates his judgement in the following way: given that he currently believes team i to be the best, he picks (at random) a game played by team i against some opponent j. With probability p, the voter moves to the state corresponding to the game’s winner; with probability (1 - p), the voter moves to the losing team’s state.

Then the transition probabilities t_{ij} from state i in the Markov chain are defined as:

$$t_{ij} = \frac{1}{N_i}\left[w_{ij}(1-p) + l_{ij}\,p\right], \qquad \text{for all } j \neq i,$$

$$t_{ii} = \frac{1}{N_i}\left[W_i\,p + L_i(1-p)\right].$$

N_i is the total number of games team i has played; w_{ij} and l_{ij} are the number of games that team i has won and lost against team j specifically; W_i and L_i are the number of games that team i has won and lost.

Given the state transition probabilities T = [t_{ij}] defined above, we then use the standard equations $\pi T = \pi$, $\sum_i \pi_i = 1$ to calculate the steady state probabilities of each team’s node. The teams are ranked in order of their steady-state probability --- the team with the highest steady-state probability is ranked first, etc.

Then we would like to find transition probabilities that answer the question, “Given that team A beat team B by x points at home (or on the road), what is the probability that A is a better team than B?”Let x(g) be the difference between the home team’s score and the visiting (road) team’s score in game g. We define r^H_x to be the probability that a team that outscores its opponent by x points at home is better than its opponent and $r^R_x = 1 - r^H_x$ to be the probability that a team that is outscored on the road by x points is better than its opponent. (Note that x can be negative to indicate that the home team lost the game.) If we denote each game by an ordered pair (i,j) of teams with the visiting team listed first, then we can write the state transition probabilities for each team i as

$$t_{ij} = \frac{1}{N_i}\left[\sum_{g=(i,j)} (1 - r^{R}_{x(g)}) + \sum_{g=(j,i)} (1 - r^{H}_{x(g)})\right],$$

$$\text{for all } j \neq i,$$

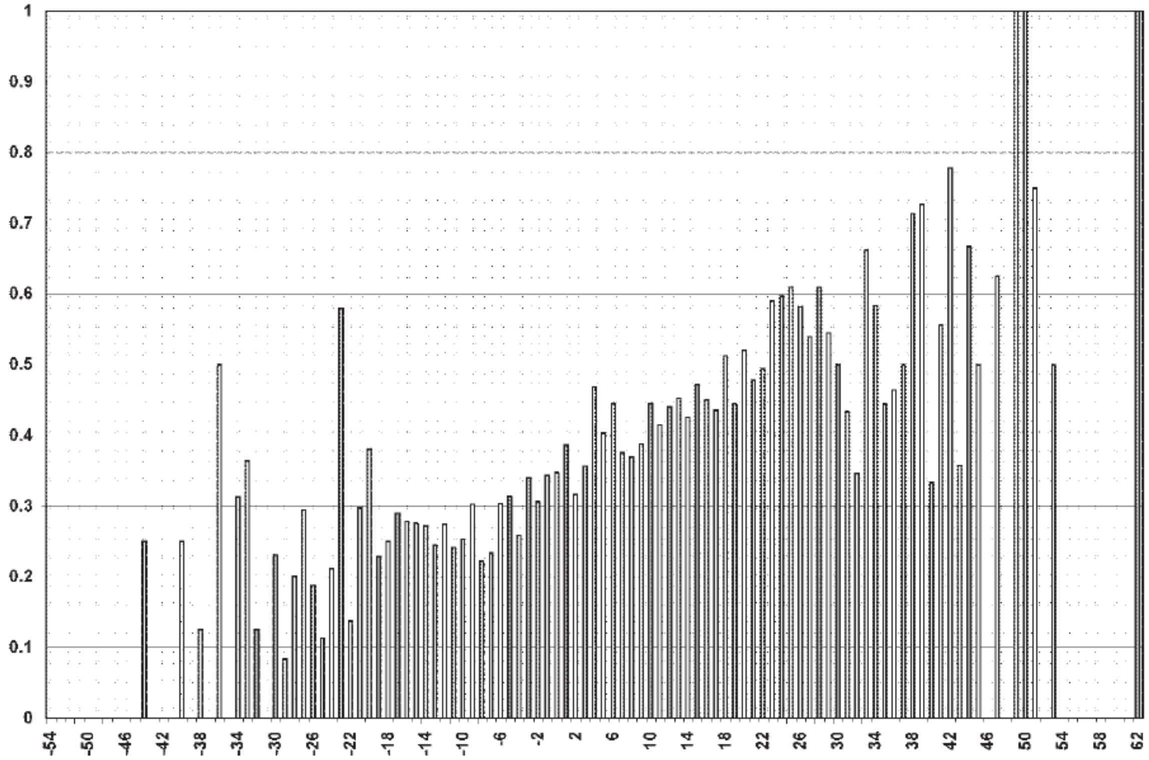
$$t_{ii} = \frac{1}{N_i}\left[\sum_{j}\sum_{g=(i,j)} r^{R}_{x(g)} + \sum_{j}\sum_{g=(j,i)} r^{H}_{x(g)}\right].$$

Then we present a logistic regression model estimating values of r^{R}_{x} and r^{H}_{x} for each x.

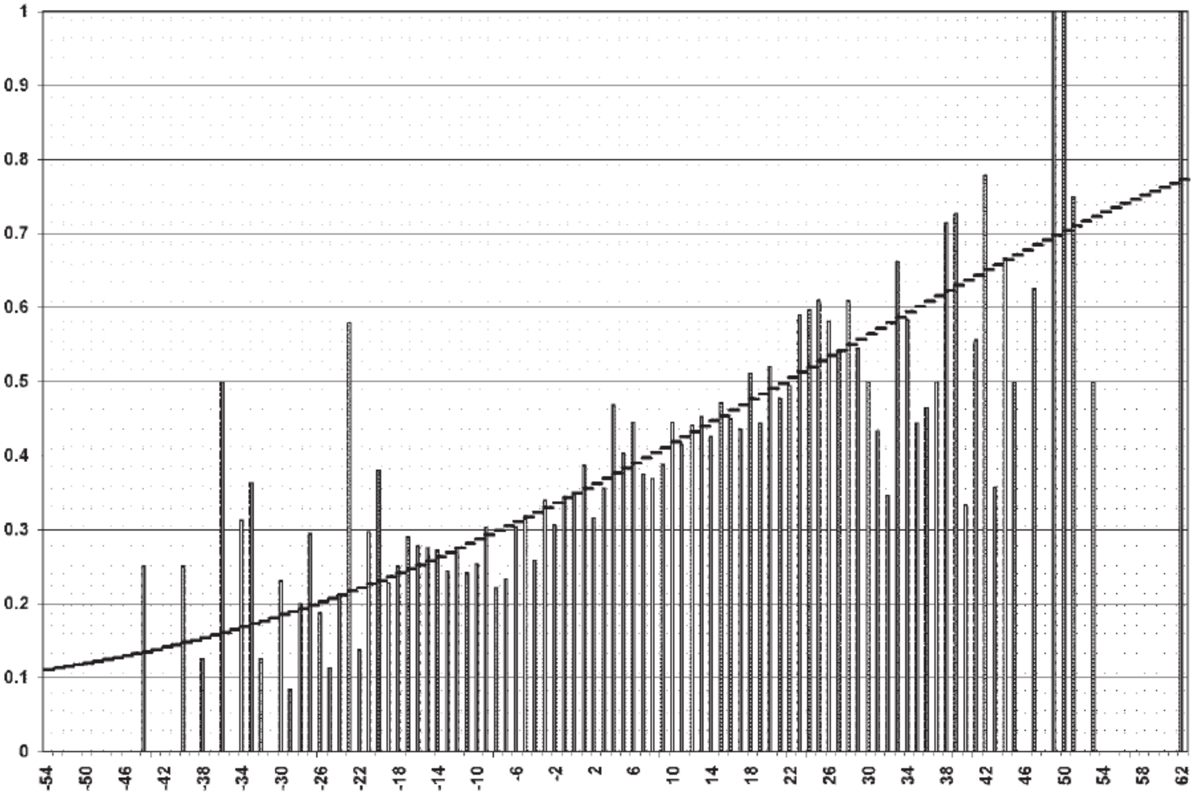
There are two steps:

1. Using home-and-home conference data, estimate an answer to the following question: “Given that team A had a margin of victory of x points at home against team B, what is the probability that team A beat team B in their other game, on the road?”
2. Given these road-win probabilities, deduce r^{H}_{x} , the probability that the home team is the better team, i.e., “Given that team A had a margin of victory of x points at home against team B, what is the probability r^{H}_{x} that team A is better than team B, i.e., that team A would beat team B on a neutral court?”

In step 1, we can calculate the probabilities results as:



Then, to obtain a better, smoother estimate of win probability, we use a logistic regression model to find a good fit. Seems as:



In step 2, we use the above results with a little modification: add a home-court advantage h. We assume that playing at home increases a team's expected point spread by h. So, the answer for step 2, in the case of a neutral-court game, a team that wins by x points at home would be expected to win by x - h at the neutral site. This gives the results of r^H_x .

After we get r^H_x , we can go on to get state transition probabilities T and steady-state probability π . And finally get the ranking for each team.

Then, we need to estimate the team-vs.-team win probabilities. Similar as above logistic regression, we can firstly calculate the team-vs.-team win probabilities by the steady-state probability π, and then smooth it and get a logistic regression model for these:

$$p_{ij} = 1 - \frac{e^{-1834.72(\pi_i - \pi_j)}}{1 + e^{-1834.72(\pi_i - \pi_j)}}.$$

Finally, the model above is used to predict all the possible team-vs.-team win probabilities in 2016.

The second approach makes use of SVMs and statistics computed from the historical data. SVMs are discriminative classifiers. They are supervised learning algorithms that are used to learn a hyperplane that is able to separate the data. Since there may be many hyperplanes that are able to separate the data the algorithm maximizes the margin the hyperplane of the training data, meaning the hyperplane that has the largest distance from data points that are closest to the boundary.

Often times the data is not linearly separable. In these cases a kernel is used in order to map the input to high-dimensional feature spaces. Some common kernels are polynomial, Gaussian radial basis function, and hyperbolic tangent.

In order to train the SVM, training data is required. The raw data needs to be processed and transformed into feature vectors and labels. The historical data is used to compute statistics for teams. Statistics from one team and their opposing team are used as input and the result from that game is used as the label. Once the SVM is trained it is applied to the new data to provide predictions.

Dataset

The data comes from the Kaggle March Machine Learning Mania 2016 competition. Various types of data were included such as regular season game results, tournament game results, tournament seeds, and some files to map the data together. The data gathered was from 1985 through to 2015.

Regular season game results and tournament game results include data such as the season, the day number of the season, teams, scores, number of overtimes, and the location of the winning team. Starting in 2003 additional information is provided. Field goals made, field goal attempts, three pointers made, three pointer attempts, free throws made, free throw attempts, offensive rebounds, defensive rebounds, assists, turnovers, steals, blocks, and personal fouls for each team were given.

Season	Daynum	Wteam	Wscore	Wloc	Numot	Wfgm	Wfga	Wfgm3	Wfga3
2003	10	1104	68	N	0	27	58	3	14

Wftm	Wfta	Wor	Wdr	Wast	Wto	Wstl	Wblk	Wpf
11	18	14	24	13	23	7	1	22

Example record without the losing team

While other information is provided, only the regular season game results and tournament game results were used to train our models. The data was transformed into an intermediary format and later turned into training data.

Evaluation

For the SVM approach we developed a number of features to be used. First the data needs to be processed. We need a way of representing a team and we are given a collection of games that were played. A potential option is to represent a team as the statistics of its games.

The composition of a team, its players and staff, are likely to be the same within a season but this may not be true over a number of seasons. It was decided that any information for a team would come from the regular season of that team for that season. With that in mind the teams became a collection of games that they played during the regular season of a specific year.

There are many values that you could calculate with the game results. With a single record of game results it is possible to calculate the field goal rate, three pointer rate, and free throw rate of success. Another thing that could be done with a single row of game data is calculating the difference between the two opposing teams.

Now that we've done we we could with a single game result we need to do something with the collection of results. Naturally we could sum up all the data but teams don't necessarily play the same number or games. Instead we use the game results to calculate the average, median, and standard deviation of score, field goals made, field goals attempted, and other fields.

Once the intermediary representations are created it is time to transform them into features and corresponding labels. The tournament game results file gives us games that take place in a neutral location which is the situation we would like to predict. Since we knew the teams that were playing and which team won we added the team statistics into the feature vector and the label became the index of the winning team.

After doing k-fold cross validation, k being the number years of data, and fiddling with features we found the performance to me just slightly better than random. Instead of just adding the raw team statistics to the feature vector the statistics of the first team were divided by the statistics of the second team. Also the statistics of the first team were divided by the statistics both teams combined. By adding more features the performance of the SVM improved slightly so we looked for other features that could be added.

One field that we didn't make use of was the location field. Games can take place in three different locations, home, away, and neutral. Only ten percent of the games in the regular season dataset were from neural locations. The home team appears to have an advantage so the data was grouped into home games, games not at home, and all games. The feature vector now consists of data from all vs all, home vs home, away vs away, home vs away, and away vs home. Following our validation scheme we ended up with a total of 362 features.

All those features helped improve our accuracy some more. Then we realized we could use neutral games from the regular season to increase the training set size. Only games that were past day number 100 were use because there were close to the end of the season and seemed more likely to be the result had it occurred during the tournament. After this the accuracy on our training set was 84% using the Gaussian radial basis function as the kernel to the SVM.

Aside from SVM we tried other machine learning algorithms using the same training data. When softmax regression was applied it appeared to get results that were comparable to SVM. We also tried using three layer neural network and it always seemed to overfit the training data so the results were not very good.

Conclusion

This is a very difficult problem. There are many unknown variables and the better team can still lose. Using only historical we can get a sense of what team is more likely to win. Combined with machine learning we can do much better than randomly guessing.

Team Roles

Originally the plan was to have Ying work on LMRC and Hoanh to work on SVM. Once those were implemented the LMRC would be used to calculate and smooth a number of probabilities which would then be used in the SVM to predict the outcomes of the games.

References

1. S. P. Kvam and J. S. Sokol. A logistic regression/Markov chain model for ncaa basketball. Naval Research Logistics, 53:788–803, 2006.
2. Zifan Shi, Sruthi Moorthy, Albrecht Zimmermann. Predicting NCAAB match outcomes using ML techniques – some results and lessons learned.2013.
3. ALEXANDER DUBBS. STATISTICS-FREE SPORTS PREDICTION. 2015.
4. Mark E. Glickman* and Jeff Sonas. Introduction to the NCAA men's basketball prediction methods issue. 2015.