
Crashing the Dance: Learning the Behavior of the NCAA Basketball Selection Committee

Yushi Jing
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332
yjing@cc.gatech.edu

Andy Cox
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332
andy@cc.gatech.edu

Abstract

We propose a machine learning approach to forecasting the behavior of the NCAA Division I Men's Basketball Committee in selecting and seeding teams for the tournament. We use several methods along with several subsets of features to achieve results comparable to human prognosticators.

1 Introduction

Each year in early March, the NCAA Division I Men's Basketball Committee (the *committee*) gathers in a hotel suite to make decisions that affect participants and supporters of college athletics across the United States: the creation of the bracket for the NCAA Division I Men's Basketball Championship (a.k.a., the Big Dance). The lucky 65 teams earn the chance to compete for the national championship.

The selection process [1] and the tournament itself are followed intensely by the public. Each year, millions of people watch on television; in 2005, 5 million people watched the tournament selection show, and an estimated 15 million people watched North Carolina defeat Illinois in the final game [2, 3, 4]. In addition, an increasing number of people (commonly known as *bracketologists*) attempt to forecast the committee's work, including Jerry Palm's CollegeRPI.com [5] and ESPN.com's Bracketology [6].

The process of predicting the bracket generated by the committee is made difficult by the number of factors they consider and the opaqueness with which they do. The committee releases publicly nothing but the final bracket; otherwise, no indication is given as to the relative importance of a team's attributes. We must therefore infer the hidden features by examining the committee's behavior in previous selections along with the attributes of teams considered for those selections. This understanding could lead to more accurate predictive systems, which could not only satiate the public's desire for information, but also help teams improve their standing in future seasons. For example, if schedule strength is an important factor, a team may try to schedule more difficult opponents the following season.

2 Background information

The process and principles of establishing the tournament bracket comprise three problems. Of the 65 teams selected for the tournament, 31 teams qualify automatically by winning their respective conference championships. The committee selects “the best available teams to fill the at-large berths” [1] for the other 34 spots. We will call this the *selection problem*. After selecting the 65 teams, the committee seeds (i.e., ranks) the teams from 1 through 65. (The terms *rank* and *seed* are often used interchangeably.) The result is often called the *seed list* or *S-Curve*, with teams shown in groups of four called *seed lines*. We will call this the *seeding problem*. The committee uses the seed list to place the teams into the championship bracket. We will call this the *bracket placement problem*. We will only consider the selection and seeding problems; the bracket placement problem is largely algorithmic based on the output of the seeding problem and uses additional principles not considered in the other problems.

2.1 Learning how to rank

In the selection and seeding problems, the committee attempts to select or rank one team over another. This leads to problems over defining what makes one team better than another, particularly when each team does not play all others. The game-to-game variance in team performance also causes difficulty in generating a global ordering.

Numerous ratings systems have been created over the years, from polls among media or coaches to systems based on mathematical models. (Jeff Sagarin’s rating system [7] is probably the most well known.) Our goal is to devise not just another ratings system based on a mathematical model, but rather one based on predicting the behavior of the committee based on its past behavior. This assumes some consistency in the principles (and application of those principles) of the committee despite its continually changing composition. Based on anecdotal evidence and other research, this is a reasonable assumption.

2.2 Ranking label limitation

The secretive nature of the committee leads to problems compiling training data for both problems. As a result, we must infer with inherent error the inner workings of the committee, including hidden features, from the committee’s public desiderata.

The committee does not publicly release their 1-65 seed list, so we have to infer a partial ordering based on the seed lines. Unfortunately, the mapping from the 1 through 65 list to the seed lines is not always exact. A team may be moved up or down one seed line from their natural seed line in order to fulfill certain criteria for the bracket. As a result, we cannot assume that the four teams on the #2 seed line were ranked 1 through 4 overall, because it is possible that one of them was exchanged with a team from the #3 seed line to fulfill bracket placement criteria. Also, even if the seed lines are exactly mapped from the 1 through 65 list, we cannot infer an ordering among the teams on a seed line. As a result, we will use the seed line values as labels for the classifiers in the seeding problem.

2.3 Redefining At-large teams

We already know which teams qualify automatically and which are at-large selections. In order to create labels for training the selection problem classifiers, we would like to know not only which teams were selected at-large, but also which teams would have been selected at-large had they not qualified automatically. This gives us more training instances, helping us build more accurate classifiers.

We can confidently assume that any automatic qualifiers with better seeds would have been

selected at-large, but we must be careful making assumptions about teams near the worst (according to the seeds) at-large selection (commonly known as the *bubble*). The actual seeds suggest which teams are among the last selected at-large, but we cannot definitively identify the last at-large selection. The problem of moving teams away from their natural seeding also complicates this; an automatic qualifier that is originally ranked behind the last at-large selection (and thus would not have been an at-large selection) may actually end up with a higher actual seed in the final bracket. We have chosen to use a combination of the actual seeds and common sense to estimate which of the automatic qualifiers we label for training data as teams that were worthy of at-large selections.

2.4 Feature space

The committee selected teams based on their performance against other NCAA Division I teams. Therefore the raw feature space is a collection of games played this season, including the game outcome and other attributes such as the game site. Division I teams play thousands of games per season (4995 games during the 2005 regular season), so using the raw feature space would suffer from both the curse of dimensionality and computational complexity. Also, it is generally easier for committee members to use aggregated features than examine (or remember) the outcome of every game.

Therefore, we have identified potential high-level features to both reduce the dimensionality of the feature space and better simulate the actual selection process. At any point in the process, committee members can request a “nitty-gritty report”, which includes many of these features. The Ratings Percentage Index (RPI), a simple formula developed by the NCAA, is at the core of most of these features. It is also widely thought, though not publicly acknowledged by the committee, that they also use other features, including Jeff Sagarin’s ratings [9]. While they may not be explicitly used, other features like rated in media or coaches polls have exhibited some correlation with the bracket. We consider:

- Rank by RPI for all games, only conference games, and only non-conference games [5]
- Rank in Sagarin ratings and coaches poll
- Record in all games, conference games, road games, last 10 games, games against various groups of teams as ordered by RPI rating. Records are normalized in most cases by using winning percentage or win-loss difference.

2.5 Related work

The closest related work we could find to the problem of automating NCAA Tournament selection is the Dance Card [8] work by Coleman and Lynch. They analyze the most significant attributes in previous at-large selections and use linear regression to generate an index for each team based on these attributes. They did not attempt to predict seeds.

Several of our techniques were inspired by existing work. Our selection of features is influenced by the results of the Dance Card work. We also used the idea from CollegeRPI.com (via the Dance Card) of pre-processing our test data to eliminate teams from at-large consideration that, based on the committee’s history, have no realistic chance of selection. For example, no team with a losing record has been selected at-large.

3 Model

For each team t_i , We associate an instantiation of the feature vector x_{t_i} with a Selection Label $y_{t_i} \in \{0, 1\}$ and a Seeding Label $S_{t_i} \in \{0, 1, \dots, 16\}$. We define $X =$

$\{x_{t_1}, x_{t_2}, \dots, x_{t_m}\}$, $Y = \{y_{t_1}, y_{t_2}, \dots, y_{t_m}\}$ and $S = \{s_{t_1}, s_{t_2}, \dots, s_{t_m}\}$ where m is the number of teams eligible for tournament consideration.

3.1 Selection problem

The selection problem is to learn a function $f(X) : R^m \times R^n \rightarrow \{0, 1\}^m$ to minimize the Hamming distance between $F(X)$ and Y , where m is the number of team candidates and n is the number of attributes in the feature vector. When the features X and labels Y are (or assumed to be) independently and identically distributed (IID), an optimal binary classifier $f_{IID}(X_{t_i}) : R^n \rightarrow \{0, 1\}$ trained on X and Y minimizes the hamming distance. Note that the feature space for f_{IID} is limited to the particular team being examined. For this reason, we can simply use a SVM or naive Bayes with MAP estimator to map the feature space of a given team into a binary label. This is our first approach, which we denote as **IID-SVM** and **IID-NB**.

However, the selection problem violates the IID assumption. For example, since there is only N number of berths available per year, the probability of a team with 15 wins receiving berth depends on the number wins other teams have. Therefore, the selection label for Duke essentially depends on all the features from all the candidate teams. This creates a feature space exponential to the number of teams and attributes per team, which is impractical with limited training data and computational resources.

One way to alleviate this problem is to have our domain knowledge assist classifier construction. Since we know that the committee selects the best N teams for the N at-large berths, Selection Problem can be converted to a Ranking Problem. Therefore, we need to construct a Scoring function f_s , a Ranking function f_r and a At-Large Selection function f_a where

$$f_s(x_{t_i}) : R^n \rightarrow R \quad (1)$$

$$f_r(f_s(x_{t_1}), \dots, f_s(x_{t_M})) : R^M \rightarrow Perm(1, \dots, M) \quad (2)$$

$$f_a(f_r) : Perm(1, \dots, M) \rightarrow \{0, 1\}^M \quad (3)$$

Since f_r and f_a are fixed¹, our task is to devise the best scoring function f_s that outputs a confidence score that minimize the hamming distance between the predicted label and Y . In our case, we simply use the margin from **IID-SVM** and the posterior from **IID-NB** as f_r . We denote this group of method as **RANK-SVM** and **RANK-NB**.

RANK-SVM and **RANK-NB** learns a ranking function with binary label Y . Since Y treats all teams in each category equally, misclassifying Duke is no different from misclassifying a bubble team, say Iowa from this year. Intuitively, seed label S should be incorporated into f_s to improve the accuracy of ranking. The simplest way is to re-weight the data based on the team seeding. We denote the new Ranking function as **RANK-NB-W**. An alternative method is to simply replace Y with S to train a supervised multi-class Bayesian network, denoted as **RANK^m-NB**. **RANK^m-NB** outputs a confidence score by taking the dot product of the multi-class posterior vector and the multi-class class label vector.

It is interesting to see how different weighting schemes in **RANK-NB-W** affect the decision boundary. If we give more weight to teams that are far way from the cutoff boundary, we can in principle find a better global ranking. However, the resulting weighted training sample will bias towards teams that are already easy to classify and will potentially blur the difference between the worst at-large team and the best left-out teams. Inversely if we apply more weights to borderline teams, the training algorithm will focus on hard-to-classify teams. We denote the first variation as **RANK-NB-W1** and the later as **RANK-NB-W2**.

¹ f_r and f_a assigns positive label to the teams with top N scores, and negative label to the rest.

3.2 Seeding problem

The seeding problem is the construction of function $f(X) : R^m \times R^n \rightarrow \{1, \dots, 16\}^m$ that minimize a scoring criteria. As with the selection problem, there are three different approaches. **IID-NB** in this case is a multi-class naive Bayes classifier with MAP estimator. **RANK-NB-W** is straightforward in this case. To convert the posterior from multi-class to a confidence score, we simply take the dot product of the posterior to the class label. For example, if a test instance is assigned a 90% probability of seed line #1 and 10% probability of seed line #2, the expected seed is $(0.9)(1) + (0.1)(2) = 1.1$. As in the selection problem, we rank the teams by the expected seed to generate the 1 through 65 seed list. All the Scoring functions are used in conjunction with Ranking function (same as previous problem) and Seed Selection Function f_{ss} is defined as

$$f_{ss}(f_r) : Perm(1, \dots, M) \rightarrow \{1, \dots, 16\}^M \quad (4)$$

We also devised a collection of pairwise scoring function **PAIRWISE-SVM**, **PAIRWISE-NB**, **PAIRWISE-SVM** and **PAIRWISE-NB-W** as alternative scoring function to **RANK-NB** and etc. Pairwise scoring function consists of a decompose function f_{d1} that breaks the features into pairwise comparison features:

$$f_{d1}(X) : R^M \times R^N \rightarrow R^{(M-1)^2} \times R^N \quad (5)$$

and a seeding decompose function f_{d2} that decompose the seeding into a collection of pairwise preference function based on the seed order.

$$f_{d2}(X) : \{1, \dots, 16\}^M \rightarrow \{0, 1\}^{(M-1)^2} \quad (6)$$

In case there is a tie, f_{d2} randomly assigns a label.

For every pair of features (from two different teams), the new feature is simply the difference of these two. Since we significantly increased the amount of training data (m to $m-1$ squared) without affecting the dimension of feature space, we believe the pairwise scoring function is superior to the standard scoring function we described in the previous section. Cohen et al. [10] proposed similar technique to devise a global rank. However, their pairwise classifier is a linear discriminant function constructed by boosting-like algorithm. In our case, we simply use naive Bayes and SVM.

Note that the input to the seeding problem is only the set of 65 teams that are either automatic qualifiers or are identified as at-large teams. In other words, we only seed the teams that we have selected for the tournament, just as the actual committee does. In fact, we cannot generate meaningful training labels for teams that were not selected, because they were not assigned a seed. It would be incorrect to assume that a team seeded #16 would be ranked higher than a team that just missed being an at-large selection, because the #16 seeds are typically from the weakest conferences and are ranked in the last quartile in most computer rankings.

4 Empirical evaluation

In the previous section, we proposed several ways to improve the prediction accuracy, including feature selection, ranking, weighting the data, and classifier selection. We will next analyze the advantages and disadvantages of these approaches. We use a combination of Weka [11] and Matlab to implement our experiments.

4.1 Measuring success

First, we define our evaluation function. We define the selection error as the percentage of at-large teams we select incorrectly. The definition of success in the seeding problem is

more complicated, so we use several measures. First, we define $Match_0$ as the percentage of seeds we predict exactly. Because a team may be moved up or down one seed line from its natural seed to satisfy bracket placement rules, we define $Match_1$ as a successful prediction of a team’s seed as one that is within one seed line of its actual seed to account for this input noise. Prognosticators such as CollegeRPI.com and ESPN.com’s Bracketology typically use this metric. Last, we define $Match_{SAD}$ (Sum of Absolute Difference) as the combined distance between the predicted and actual seeding. $Match_{SAD}$ gives us a combined measure prediction accuracy. The worst value for $Match_{SAD}$, assuming that all at-large selections are correct, is 512. We will not assign $Match_{SAD}$ values for mis-selected teams, assuming that the seed and selection results will be used together. Unless otherwise noted, all experiments use the season prior to the test season as training data.

4.2 Feature selection

We define three sets of features. First, we rank all the features according to their information gain and select the best set (excluding dependent features) from the total of 37 features (13 here). We call this set $Feature_M$. We then manually select 5 features (mostly the team ratings) from $Feature_M$ and call this set $Feature_S$. These features are widely considered as the most important features for bracket consideration. We also define $Feature_L$ to include the entire 37 features. The $\{S, M, L\}$ refer to the relative size (small, medium, large) of the feature set. Table 1 compares the features sets for several methods.

Feature set	RANK-NB			PAIR-NB		
	$Match_0$	$Match_1$	$Match_{SAD}$	$Match_0$	$Match_1$	$Match_{SAD}$
$Feature_S$	38.1%	73.8%	68.5	35.4%	70.0%	79.0
$Feature_M$	39.6%	73.8%	69.5	38.5%	76.5%	63.0
$Feature_L$	30.2%	68.5%	85.0	43.5%	77.4%	60.2

Table 1: The average accuracy of **RANK-NB** and **PAIR-NB** on the seeding problem for 2001-2004. The best in each category is shown in bold. $Feature_S$ and $Feature_M$ perform better than $Feature_L$ in **RANK-NB** and its variations due to insufficient training data. **PAIRWISE-NB** benefits from a richer feature set.

4.3 Method selection

Methods	$Match_0$	$Match_1$	$Match_{SAD}$
IID-NB $Feature_M$	15.4%	43.8%	89.0
RANK-NB + $Feature_M$	39.6%	73.8%	69.5
RANK-NB-W2 + $Feature_M$	39.1%	74.6%	67.8
PAIR-NB + $Feature_L$	43.5%	77.4%	60.2
PAIR-NB-W1 + $Feature_M$	42.0%	77.4%	60.0
PAIR-NB-W2 + $Feature_M$	43.5%	79.8%	57.6
PAIR-SVM + $Feature_L$	42.0%	78.0%	55.0

Table 2: The accuracy for seeding problem using different methods. The best in each category is shown in bold.

Table 2 compares the testing error for Selection problem using different methods with their best feature sets. It shows that **RANK-NB** outperforms **IID-NB** in both selection problem and seeding problem. **PAIR-NB-W2** outperforms **PAIR-NB** and **PAIR-NB-W1** by assigning higher weights to pairwise comparison that involves two similar teams. This forces Naive Bayes to focus on correctly classifying boundary teams.

We also compared linear SVM with NB ². In this experiment, **PAIR-SVM** has similar $Match_0$ errors as NB, but outperforms **PAIR-NB** on the other error measurements. Since the classification accuracy of naive Bayes relies on its attributes to be conditionally independent, SVM outperforms naive Bayes, especially with a larger feature set.

Methods	$Match_0$	$Match_1$	$Match_{SAD}$
RANK-NB + $Feature_S$	47.3%	81.9%	52.5
RANK-NB + $Feature_M$	50.8%	81.0%	52.5
PAIR-NB-W2 + $Feature_M$	43.5%	81.0%	56.0
PAIR-SVM + $Feature_L$	43.0%	82.2%	53.0

Table 3: The accuracy of for Seeding problem using different methods. In this case, we are combining training data from all seasons previous to the testing data. The best in each category is shown in bold.

Table 3 shows that combining multiple season of data significantly improves the accuracy of **RANK-NB**+ $Feature_S$ and **RANK-NB**+ $Feature_M$. It is not a very surprising result since more training data leads to less overfitting. However, it does suggest that selection criteria are have been fairly consistent over the past 5 years.

4.4 Forecasting the 2005 bracket

Method	False Positive	False Negative
Sagarin	Maryland, Tex. A&M, Drexel, Vanderbilt	Stanford, UCLA, UAB, St. Mary's
RPI	Miami-OH, Wichita State, Buffalo	Iowa State, NC State, UAB
RANK-NB + $Feature_S$	Miami-OH, Wichita State	NC State, Iowa State
RANK-NB + $Feature_M$	Miami-OH, Wichita State, Tex. A&M	NC State, Iowa State, St. Mary's
RANK-NB-m + $Feature_S$	Wichita State	Iowa State
CollegeRPI.com	Notre Dame	UAB

Table 4: Selection results for 2005

Methods	$Match_0$	$Match_1$	$Match_{SAD}$
RANK-NB + $Feature_S$	55.4%	78.5%	48.0
RANK-NB + $Feature_M$	50.8%	86.2%	44.0
PAIR-NB-W2 + $Feature_M$	40.0%	80.0%	58.0
PAIR-SVM + $Feature_L$	43.1%	81.5%	51.0
CollegeRPI.com	63%	83%	36

Table 5: Seeding accuracy on 2005 data trained on combined dataset from 2000 to 2004. The best in each category is shown in bold.

Tables 4 and 5 show our results compared to that of the RPI and Sagarin rankings and a leading human bracketologist (CollegeRPI.com) in selecting at-large teams and seeding. **RANK-NB-m** is the best of our selection methods, missing only one team. The next best is **RANK-NB**+ $Feature_S$. Our method is also competitive with human prediction.

The **RANK-NB** methods outperformed the pairwise methods on the 2005 data. This seems to indicate that the committee based their decisions on a relatively small number of features, mainly ranking features including Sagarin and RPI. These features are already the result of pairwise comparison. This explains why methods excel in larger feature space did not

²Our earlier results showed that SVM used in conjunction with RBF or Polynomial kernel with standard parameter severely over-fit the data.

Testing Years:	2001	2002	2003	2004	2005
SAD error with RANK-NB+ <i>features</i>	84	72	64	68	42

Table 6: The testing error for **RANK-NB** on Small feature set.

perform as well as more simpler method in 2005 data. Table 6 seems to indicate that the committee is increasingly relying on ranking features instead of raw results, though the small sample sizes could be deceiving.

5 Conclusion and future work

We have seen that the small amount of training data for the selection and seeding problems makes it difficult to conclude a priori that one method will perform better than another. A method that does well on 2005 data may, like a human forecaster, perform badly on 2006 data. Future work should investigate whether multiple methods could be combined using ensemble or boosting methods.

This is a real-world machine-learning problem, so there is no absolute solution. Our results have indicated that lack of data means that a simpler classifier probably works better. Unfortunately, like fans of college basketball, we must wait until the next selection committee assembles to generate the next test set.

References

- [1] NCAA Division I Men’s Basketball Committee. NCAA Division I Men’s Basketball Championship principles and procedures for establishing the bracket (last accessed March 30, 2005). <http://www.ncaasports.com/basketball/mens/story/6985142>.
- [2] Nielsen Media Research. Top ten primetime broadcast TV programs (last accessed April 6, 2005). http://www.nielsenmedia.com/ratings/broadcast_programs.html.
- [3] CBS Television. CBS Sports’ coverage of NCAA basketball championship weekend and “Selection Show” Sunday earns high marks (last accessed April 6, 2005). <http://www.viacom.com/press.tin?ixPressRelease=80704183>.
- [4] Associated Press. Ratings for NCAA title game best since 1999 (last accessed April 6, 2005). http://www.usatoday.com/sports/college/mensbasketball/tourney05/2005-04-05-tv-ratings_x.htm.
- [5] Jerry Palm. CollegeRPI.com (last accessed March 30, 2005). <http://www.collegerpi.com/>.
- [6] ESPN.com. Bracketology (last accessed March 30, 2005). <http://sports.espn.go.com/ncb/bracketology>.
- [7] Jeff Sagarin. Jeff Sagarin’s college basketball ratings (accessed March 14, 2005). <http://www.kiva.net/jsagarin/sports/cbsend.htm>.
- [8] B. Jay Coleman and Allen K. Lynch. Identifying the NCAA Tournament “dance card”. *Interfaces*, 31(3), May-June 2001.
- [9] Jeff Sagarin. Meet Jeff Sagarin (last accessed April 6, 2005). <http://www.kiva.net/jsagarin/p2wjjeff.htm>.
- [10] W.W. Cohen, R. E. Schapire, and Y. Singer. Learning to order things. In *Advances in Neural Information Processing Systems*, volume 10. The MIT Press, 1998.
- [11] Ian H. Witten and Eibe Frank. *Data Mining: Practical machine learning tools with Java implementations*. Morgan Kaufmann, 2000.