Full Report
March Madness
Juan-David Dominguez & Michael Gofron
juandominguez2017@u.northwestern.edu &
michaelgofron2017@u.northwestern.edu
EECS 349 Machine Learning, Northwestern University

We were motivated to find a way to generate a perfect bracket or a better bracket than average based on the idea that it is a mathematically interesting task and economically advantageous. It is mathematically interesting because the predicted odds of getting a perfect bracket are 1 in 1,610,543,269. These odds are so slim because one would have to guess the correct outcome of 63 consecutive games, where an early mistake guarantees the rest of the bracket is incorrect. Additionally, it is economically advantageous because we can make money by betting with our friends and others on who has a better bracket and there is usually money offered for a perfect bracket by multiple companies.

We are using the classifiers from the python Scikit library. This library includes our current main classifier Decision Tree. We took in many features of a team that was compiled throughout the season games*.

These team statistics include a team's performance and their opponents' collective efforts. For example, team statistics would include field goal points made, field goal points attempted and field goal percentage, as well as opponent field goals made, opponent field goal points and opponent field goal percentage, among many others. We also manipulated each feature (except for percentages) to be per game to prevent skewing from the number of games a team could play in the whole season, since some teams played more games than other teams. We then took the differential between a team's proper statistic and its opponent statistic. In all, we took in 120 features for a team into our decision tree as can be seen in Appendix A. For training, a game would be the instance where we would pit team A's statistics vs team B's statistics and then whether team A won was the outcome.

We chose decision tree for multiple reasons. From a human-perspective, the way in which decision trees print in "if-then" routes where the early splits show which attributes give more information gain allows for us to gain a better intuitive understanding for the learning problem in general and helps us prepare for using other classifiers in the future. Furthermore, the problem tends towards decision tree. The discrete, binary output space works well with decision trees. The instances are clearly labeled making it supervised learning and the instances are represented by attribute-value pairs, which makes this learning problem well fitted for a decision tree. Lastly, decision trees are very robust to noise and this is very helpful since we are using a large data set. This robustness is contrasted with random forest.

We experimented with using the Random Forest as a classifier but the results were in general extremely subpar to those of the decision tree. We believe it deals with the size and relevance of our data as well as the nature of Random Forest. Random Forests breaks the data into multiple parts and then creates a small tree for each part of the data. Since there are so many more teams in the regular season games than there are in the NCAA tournament, there are exponentially more regular

Juan David Dominguez 6/10/15 6:18 PM
Formatted: Indent: First line: 0.5"

Juan David Dominguez 6/10/15 5:43 PM
Deleted: (for the year 2015).

Juan David Dominguez 6/10/15 5:46 PM
Deleted: . Also,

Juan David Dominguez 6/10/15 6:18 PM
Deleted:

Juan David Dominguez 6/10/15 5:47 PM
Deleted: ed

Juan David Dominguez 6/10/15 5:48 PM
Deleted: a

Juan David Dominguez 6/10/15 5:48 PM
Deleted: from the Scikit library

Juan David Dominguez 6/10/15 5:49 PM
Deleted:

Juan David Dominguez 6/10/15 5:49 PM
Deleted: d

season games, especially more of lower caliber teams. These games and teams are more irrelevant to the NCAA tournament and when the Random Forest breaks the data into smaller parts, these games will dominate the data set—to the point where there might not be a single game played by an NCAA tournament contender. This hypothesis is corroborated when we iteratively trained on the results of all other tournaments' data and verified with the one tournament that was not trained on. The random forests did comparatively better than a decision tree. Nonetheless, we choose decision tree in order to hopefully boost it in the future and use the intuitive understanding it gives to help us stack it with other classifiers.

The results are very interesting and further work needs to be done. The more exceptional result is that differential statistics are the most important features and, in particular Point differential is the most important feature. This seems to be in line with professional basketball analysts' opinions. Our current accuracy is on ranges from [] and is on average [78%?]. The accuracy is better for tournaments that have been deemed to have more "upsets."

Suggestions for Future Work

We would like to try using more algorithms to see the validity of other algorithms and their impact. We also want to stack algorithms on top of one other in order to produce a more effective general algorithm for the future. Additionally, there should be more work done in separating the dataset between the playoffs and the regular season for the team statistics to ensure statistics are not skewed based on the number of games a team has played in a season. We would like to make the bracket making process more automatic.

Juan David Dominguez 6/10/15 6:21 PM
Deleted:

Juan David Dominguez 6/10/15 6:30 PM
Deleted: iii. We trained our decision tree with the team stats over the season* as our features and whether a team won a game as the output space and restricting our training set to the regular season. Then we measured our success originally based on the percentage of games that the algorithm guessed correctly. ... [1]

Random Forest vs Decision Tree Training on previous playoffs and testing on a playoff

```
Performance of Decision Tree:  0.6507936507936
Checking playoffs vs Playoff
Excluding year 2010
Performance of Decision Tree:  0.52380952381
Performance of Random Forest:  0.650793650794
Excluding year 2011
Performance of Decision Tree:  0.587301587302
Performance of Random Forest:  0.634920634921
Excluding year 2012
Performance of Decision Tree:  0.714285714286
Performance of Random Forest:  0.650793650794
Excluding year 2013
Performance of Decision Tree:  0.666666666667
Performance of Random Forest:  0.634920634921
Excluding year 2014
Performance of Decision Tree:  0.571428571429
Performance of Random Forest:  0.650793650794
Excluding year 2015
Performance of Decision Tree:  0.555555555556
Performance of Random Forest:  0.746031746032
>>>
```