To enable screen reader support, press shortcut Ctrl+Alt+Z. To learn about keyboard shortcuts, press shortcut Ctrl+slash.

Toggle screen reader support

Performance Tracers (debug users only)

```
     10.719 Start       Reading document model in edit action
   0 10.719 Done   0 ms Reading document model in edit action

     10.731 Start       Calling handleResponse: SoySauceChunkedTemplateResponseHandler
   0 10.731 Start       |  [SoySauceChunkedTemplateResponseHandler#write] Render for kix.shared.soy.editBegin
```

```
  0 10.731 Start       |  |  [SoyTemplate#getInjectedValues] getInjectedValues for kix.shared.soy.editBegin
  4 10.735 Start       |  |  |  [SoyTemplate#getInjectedValue] flagDataJson
  0 10.735 Start       |  |  |  |  Injecting @interface com.google.apps.docs.flags.ClientFlagMap
  1 10.736 Start       |  |  |  |  |  Injecting @interface com.google.apps.docs.flags.ClientFlagMap[docs_oogt]
 15 10.751 Done  15 ms |  |  |  |  |  Injecting @interface com.google.apps.docs.flags.ClientFlagMap[docs_oogt]
  6 10.757 Done  22 ms |  |  |  |  Injecting @interface com.google.apps.docs.flags.ClientFlagMap
  1 10.758 Done  23 ms |  |  |  [SoyTemplate#getInjectedValue] flagDataJson
  9 10.767 Done  36 ms |  |  [SoyTemplate#getInjectedValues] getInjectedValues for kix.shared.soy.editBegin
  3 10.770 Done  39 ms |  [SoySauceChunkedTemplateResponseHandler#write] Render for kix.shared.soy.editBegin
  2 10.772 Done  41 ms Calling handleResponse: SoySauceChunkedTemplateResponseHandler
TOTAL SoyTemplate#getInjectedValue 192 (35 ms)
TOTAL DocosClientConfig 7 (0 ms)
TOTAL SoyTemplate#getInjectedValues 6 (36 ms)
TOTAL SoySauceChunkedTemplateResponseHandler#write 6 (41 ms)
```

Predicting the NCAA Basketball Tournament with ML

Andrew Levandoski and Jonathan Lobo
03.14.2017
CS 2750: Machine Learning
Dr. Kovashka

Overview

Every year, millions of college basketball fans attempt to predict the outcome of the NCAA Men's Basketball Tournament, known as "March Madness." The tournament consists of 68 teams and seven rounds of single-elimination basketball. While nobody has ever predicted the correct outcome of all 67 games held during the tournament, the emergence of more accurate machine learning techniques as well as incentives such as bracket pools and Kaggle's machine learning bracket competition have led to increased prediction accuracy based on algorithms using historical data.

Goals

1. Build a model to predict game outcomes and the probabilities of each outcome
2. Train the model on historical data from past tournaments
3. Validate the model and evaluate performance by making predictions based on historical data from years withheld from the training set
4. Use the model to predict the outcomes of games in the 2017 tournament
5. Evaluate the model's performance:
    a. In terms of accuracy
    b. Against other models

Previous Work

Inspiration for this project stems from FiveThirtyEight.com's prediction platform, an extremely popular resource for bracket construction. The platform offers both predictions and likelihoods of all potential matchup outcomes.

Further interest is derived from Kaggle.com's annual March Machine Learning Mania competition, which collects models and determines a winner based on the prediction accuracy.

The winner of Kaggle's March Machine Learning Mania 2016 used logarithmic regression and random forests, a supervised learning method in which multiple decision trees are constructed at training time and the output class is the mean prediction of the individual trees. Random forests can be used for dimensionality reduction.

Some other models used before are K-nearest neighbors, logistic regression, and neural networks.

Data Collection

To train and test our model, we will used team-level historical data provided by Kaggle (player-level data could potentially be introduced in further explorations of our model). The data is available online at the following link:

is available online at the following link:
https://www.kaggle.com/c/march-machine-learning-mania-2017/data

The data are formatted as .csv files organized as follows:

Teams

This file identifies the different college teams present in the dataset. Each team has a 4 digit id number.

Seasons

This file identifies the different seasons included in the historical data, along with certain season-level properties.

- "season" - indicates the year in which the tournament was played
- "dayzero" - tells you the date corresponding to daynum=0 during that season.
- "regionW/X/Y/Z"

RegularSeasonCompactResults

This file identifies the game-by-game results for 32 seasons of historical data, from 1985 to 2015. Each year, it includes all games played from daynum 0 through 132 (which by definition is "Selection Sunday," the day that tournament pairings are announced). Each row in the file represents a single game played.

- "season"
- "daynum"
- "wteam" - this identifies the id number of the team that won the game.
- "wscore" - this identifies the number of points scored by the winning team.
- "lteam" - this identifies the id number of the team that lost the game.
- "lscore" - this identifies the number of points scored by the losing team.
- "numot" - this indicates the number of overtime periods in the game, an integer 0 or higher.
- "wloc" - this identifies the "location" of the winning team.

RegularSeasonDetailedResults

This file is a more detailed set of game results, covering seasons 2003-2016. This includes team-level total statistics for each game (total field goals attempted, offensive rebounds, etc.) The column names should be self-explanatory to basketball fans (as above, "w" or "l" refers to the winning or losing team):

- wfgm - field goals made
- wfga - field goals attempted
- wfgm3 - three pointers made
- wfga3 - three pointers attempted
- wftm - free throws made
- wfta - free throws attempted
- wor - offensive rebounds
- wdr - defensive rebounds
- wast - assists
- wto - turnovers
- wstl - steals
- wblk - blocks
- wpf - personal fouls

TourneyCompactResults

TourneyCompactResults

This file identifies the game-by-game NCAA tournament results for all seasons of historical data. The data is formatted exactly like the regular_season_compact_results.csv data.

TourneyDetailedResults

This file contains the more detailed results for tournament games from 2003 onward.

TourneySeeds

This file identifies the seeds for all teams in each NCAA tournament, for all seasons of historical data. Thus, there are between 64-68 rows for each year, depending on the bracket structure.

- "season" - the year.
- "seed" - the seed.
- "team" - this identifies the id number of the team, as specified in the teams.csv file.

TourneySlots

This file identifies the mechanism by which teams are paired against each other, depending upon their seeds.

- "season" - the year.
- "slot" - this uniquely identifies one of the tournament games.
- "strongseed" - this indicates the expected stronger-seeded team that plays in this game.
- "weakseed" - this indicates the expected weaker-seeded team that plays in this game.

Specifications

The goal of our project will be to determine which learning model provides the best classification accuracy in predicting the outcome of any NCAA Men’s Basketball Tournament game based on selected features from Kaggle’s March Machine Learning Mania dataset and to build upon our chosen model to achieve as high an accuracy as possible. Models we will test include linear regression, logistic regression, SVM, neural network, k-nearest neighbors, and random forest.

Evaluation

Success in predicting the results of games in NCAA basketball tournament comes in two forms: accuracy in predicting the outcome of games (i.e., the percentage of games whose outcomes were correctly predicted) and points-based prediction for pool competitions in which later-round games carry greater weight. While the two tasks are correlated, maximizing success in both will not necessarily involve identical picks.

To simplify evaluation, we will weight all games equally, regardless of the round. To evaluate our model against other prediction algorithms, we must use a log loss:

$$LogLoss = -\frac{1}{n}\sum_{i=1}^{n}[y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i)]$$

,

where n is the number of games played, $\hat{y}_i$ is the predicted probability of team 1 beating team 2, and y_i is 1 if team 1 wins, 0 if team 2 wins.

A smaller log loss is better. For our purposes, games which are not played are ignored in the scoring. Play-in games are also ignored, so only the games among the final 64 teams are scored. The use of the logarithm provides extreme punishments for being both confident and wrong. In the worst possible case, a prediction that something is true when it is actually false will add an infinite value to the error score; to prevent this, predictions must be bounded away from the extremes by a small value. We can compare this score to the score of other algorithms to evaluate our performance.

Relevant Links

NCAA Basketball Tournament Wikipedia Article:
https://en.wikipedia.org/wiki/NCAA_Division_I_Men%27s_Basketball_Tournament

FiveThirtyEight.com’s 2017 Prediction Engine:
https://projects.fivethirtyeight.com/2017-march-madness-predictions/

Kaggle’s March Machine Learning Mania Competition:
https://www.kaggle.com/c/march-machine-learning-mania-2017