March Machine Learning Mania, 1st Place Winner's Interview: Andrew Landgraf

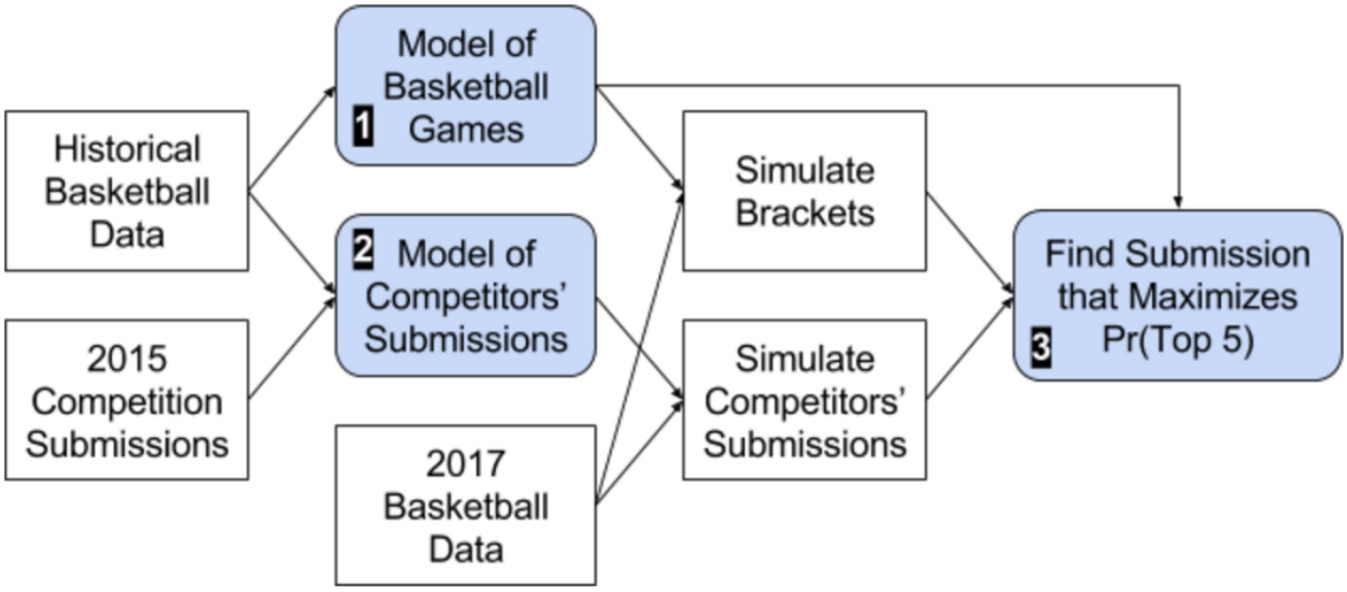
Kaggle Team (http://blog.kaggle.com/author/kaggleteam/) | 05.19.2017

Kaggle's 2017 March Machine Learning Mania competition (https://www.kaggle.com/c/march-machine-learning-mania-2017#description) challenged Kagglers to do what millions of sports fans do every year–try to predict the winners and losers of the US men's college basketball tournament. In this winner's interview, 1st place winner, Andrew Landgraf, describes how he cleverly analyzed his competition to optimize his luck.

What made you decide to enter this competition?

I am interested in sports analytics and have followed the previous competitions on Kaggle. Reading last year's winner's interview (http://blog.kaggle.com/2016/05/10/march-machine-learning-mania-2016-winners-interview-1st-place-miguel-alomar/), I realized that luck is a major component of winning this competition, just like all brackets. I wanted to see if there was a way of maximizing my luck. For example, when entering an office pool, your strategy depends on whether you are facing 5 Duke alumns or the entire office. My goal was to systematically optimize my submissions against the competition.

This competition is unique among Kaggle contests in that there is a history of submissions from previous years. My idea was to model not only the probability of each team winning each game, but also the competitors' submissions. Combining these models, I searched for the submission with the highest chance of finishing with a prize (top 5 on the leaderboard). A schematic of my approach is below. The three main processes are shaded in blue: (1) A model of the probability of winning each game, (2) a model of what the competitors are likely to submit, and (3) an optimization of my submission based on these two models.



While I believe this approach is generally worthwhile, a much simpler approach would have also won the competition, as discussed at the end.

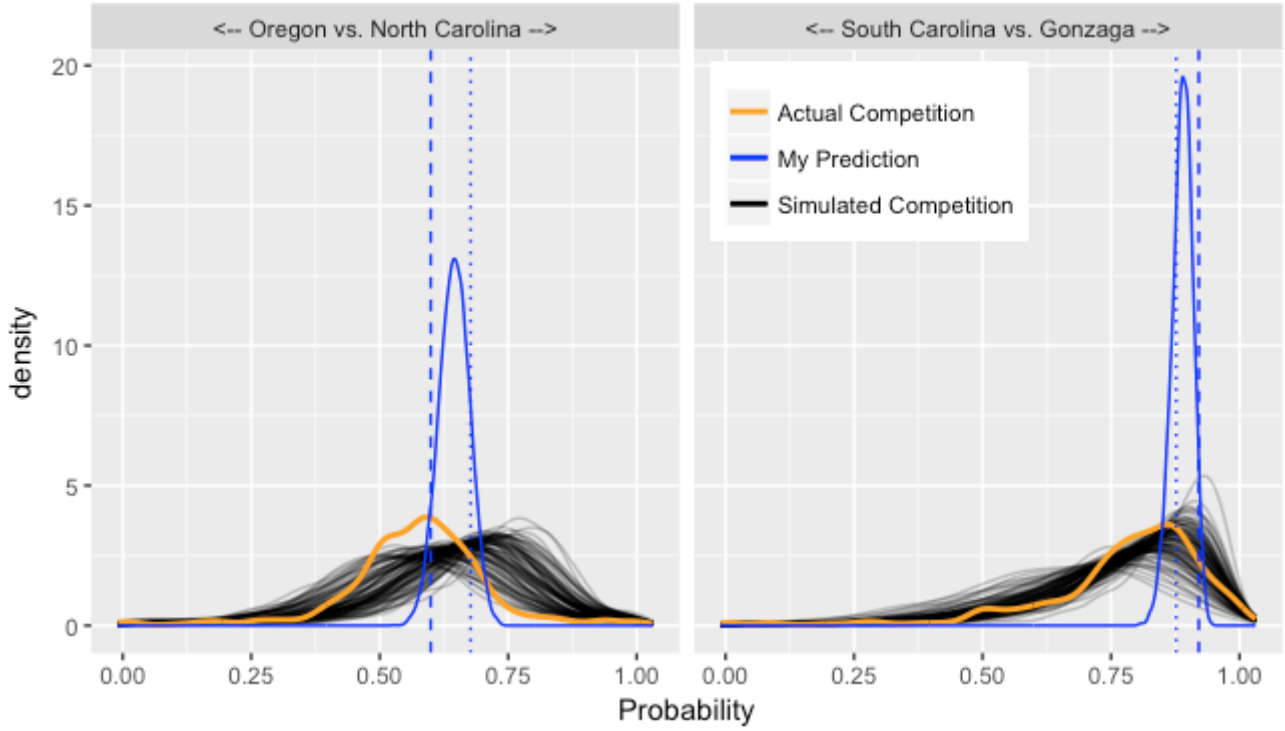
What was your approach? Did past March Mania competitions inform your winning strategy?

I kept my models simple and probabilistic. To model the outcomes of each game, I used a similar method as previous winners, One Shining MGF (https://arxiv.org/abs/1412.0248). I created my own team efficiency ratings using a regression model so that I could calculate the historical ratings before the tournament started. The ratings, and a distance from home metric (more on this later), were used as covariates in a Bayesian logistic regression model (using the rstanarm (https://cran.r-project.org/web/packages/rstanarm/index.html) package) to predict the outcomes of each game.

To model competitors' submissions, I built a mixed effects model (with lme4 (https://cran.r-project.org/web/packages/lme4/index.html)) using data from the previous competitions. I used the logit of the submitted probability as the response, the team efficiencies as fixed effects, random intercepts for competitors and games, and random efficiency slopes for competitors. I guessed that there would be 500 competitors and that 400 of them would make 2 submissions, which wasn't too far off.

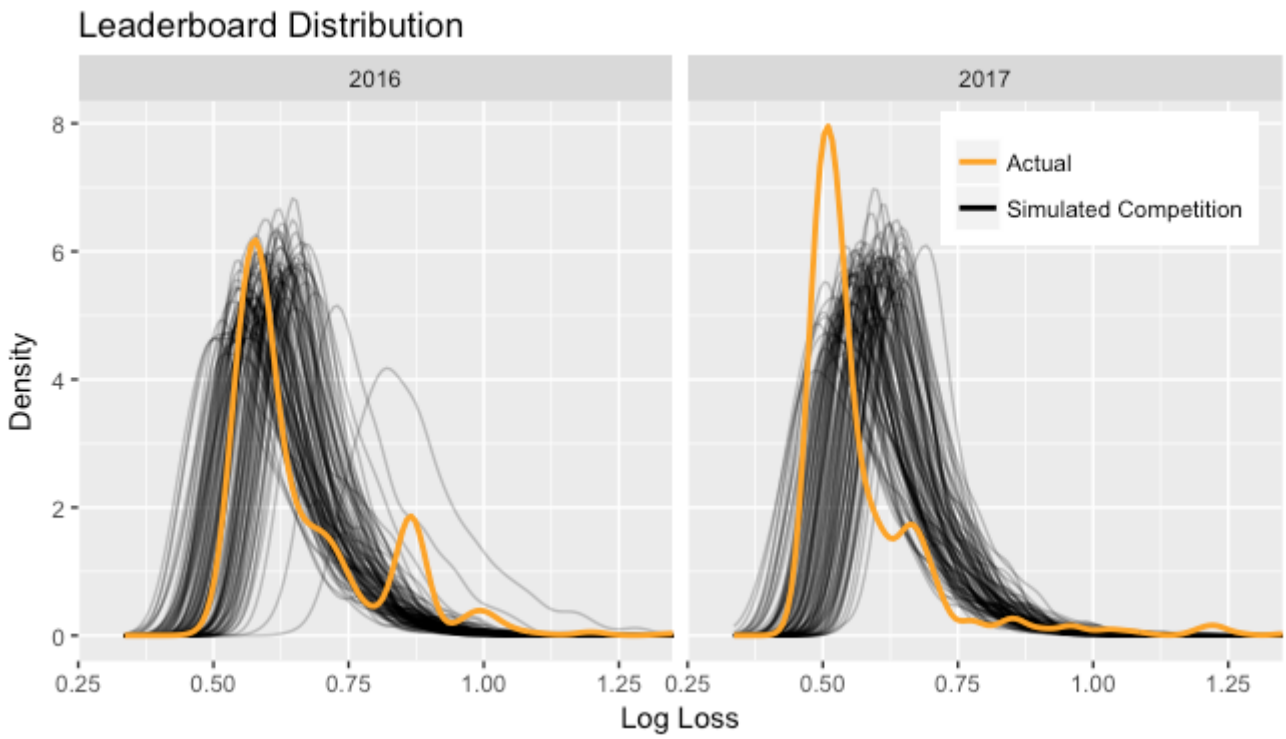
The plot below shows the models for the two Final Four semi-final games. The black lines are densities of 100 simulations from the mixed effects model and the orange line is the true distribution of competitors' predictions. They line up well for the SC vs. Gonzaga game and a little less so for the

Oregon vs. UNC game. The posterior distribution from my model is much tighter than distributions from the competitors. My two submissions are the two vertical lines.



Finally, I used these models to come up with an optimal submission by simulating the bracket and the competitions' submissions 10,000 times. This essentially gave me 10,000 simulated leaderboards of the competitors and my goal was to find the submission that most frequently showed up in the top 5 of the leaderboard. I tried to use a general-purpose optimizer, but it was very slow and it gave poor results. Instead, I sampled pairs of probabilities from the posterior many times, and chose the pair that was in the top 5 the most times. If I had naively used the posterior mean as a submission, my estimated probability of being in the top 5 would have been 15%, while my estimated probability of for the optimized submission (with two entries) went up to 25%.

The competitors' submission model was trained on 2015 data. To assess the quality of the model, I have plotted the simulated distribution of the leaderboard losses for 2016 and 2017 and compared to the actual leaderboards. 2016 seems well in line, but 2017 had more submissions with lower losses than predicted. For both years, the actual 5th place loss was right in line with what was expected.



Looking back, what would you do differently now?

A common strategy for this competition is to use the same predictions in both submissions except for the championship game, in which each team is given a 100% chance of winning in one of the submissions, guaranteeing that one of the two submissions will get the last game exactly correct. While I was aware of this strategy beforehand, I didn't realize how good it is. If I had used this strategy, my estimated probability of being in the top 5 was 27%, 2 percentage points higher than my submission. This submission would have also won the competition.

What have you taken away from this competition?

Sometimes it's better to be lucky than good. The location data that I used had a coding error (https://www.kaggle.com/c/march-machine-learning-mania-2017/discussion/28375#169431) in it. South Carolina's Sweet Sixteen and Elite Eight games were coded as being in Greenville, SC instead of New York City. The led me to give them higher odds than most others, which helped me since they won. It is hard to say what the optimizer would have selected (and how it affected others' models), but there is a good chance I would have finished in 2nd place or worse if the correct locations had been used.

Bio

Andrew Landgraf is a research statistician at Battelle (https://www.battelle.org/). He received his Ph.D. in statistics from the Ohio State University, researching dimensionality reduction of binary (https://cran.r-project.org/web/packages/logisticPCA/index.html) and count data. At Battelle, he applies his statistical and machine learning expertise to problems in public health, cyber security, and transportation.

MARCH MANIA (HTTP://BLOG.KAGGLE.COM/TAG/MARCH-MANIA/)

SPORTS ANALYTICS (HTTP://BLOG.KAGGLE.COM/TAG/SPORTS-ANALYTICS/)

10 Comments No Free Hunch 1 Login

Recommend 12 Share Sort by Best

Join the discussion…

LOG IN WITH OR SIGN UP WITH DISQUS

Name

Kirtane • 9 months ago

Thanks for sharing and congratulations. Do you think that a GAN would be able to simulate this data as well and/or is that something you tried?

2 • Reply • Share ›

andland ➚ Kirtane • 9 months ago

The similarity to GANs is interesting. The competitors' submissions do not really follow a normal distribution, so I could see GANs being able to improve their simulations.

• Reply • Share ›

Kawaii style ➚ andland • 5 months ago

I wanted to see if there was a way of maximizing my luck. For example, when entering an office pool, your strategy depends on whether you are facing 5 Duke alumns or the entire office. My goal was to systematically optimize my submissions against the competition.

1 • Reply • Share ›

Joyson Pereira • 8 months ago

Great.

Reply • Share ›

s v p • 9 months ago

Hi,

You mention " distance from home metric (more on this later)" , but you don't mention it later (unless, you are referring to an article that you intend to publish at a later today).

Reply • Share ›

andland ➚ s v p • 8 months ago

I am referring to the discussion in the last question.

Reply • Share ›

Jing Lu • 9 months ago

What package do you use for all those simulations?

Reply • Share ›

andland ➚ Jing Lu • 8 months ago

For the posterior probability simulations, I used rstanarm. For the competition submissions, I coded the simulation using the lme4 parameters.

Reply • Share ›

Pete Gordon • 9 months ago

Great Job! OH-IO

Reply • Share ›

Rakesh Kumar Dondapati • 8 months ago

Machine learning is closely related to computational statistics,which also focuses on prediction making through the use of computers.
Thanks for sharing the valuable information..!!!
we are conducting Live Free Webinar on Big data-spark and scala: https://goo.gl/Q3reSf.