Final Project: Bayesian analysis on

LEAGUE OF LEGENDS

Leonardo Placidi 1761588

Summary

Data Collection

Data exploration

The idea

The models

The best model

The diagnostic

Data Collection

1. I retrieved balanced data from the API of RIOT for League of Legends Ranked Solo/Duo games.
2. I used Python
3. Data were distributed among the 6 most played ranks (99% players)
4. Data were in csv format
5. Let’s jump fast to the code!

Data Exploration

01 I examined every feature plot (based on num.obs) and decided which were the most significant: wins, leaguePoints, tier

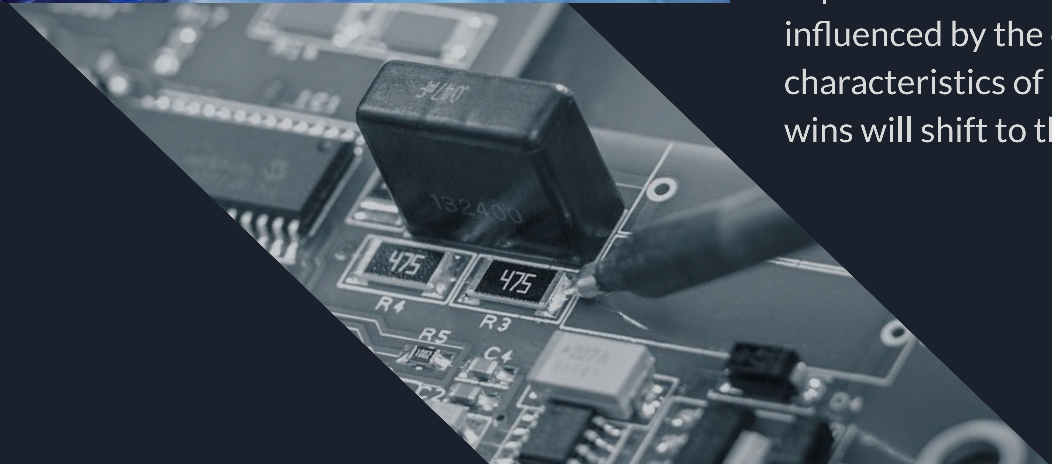
02 I showed relations between the features such as an increase of wins in higher tiers (best players play more) and more wins to achieve less LP in higher tiers.

03 I showed that if we want to model the wins, we should assume a likelihood that will change signficantly by tier, since the plots differed in width.

The idea

We want to know if we can predict the number of wins that a player must do to reach a certain tier and leaguePoints, so the model assumed priors and likelihood justified by the exploratory data analysis.

In particular the idea is that the likelihood is strongly influenced by the data exploration and the unique characteristics of every tier(to say it fast the density of wins will shift to the right as we increase tier).



The models

I assumed different likelihoods for the pr(wins|LP, tier) and I tested the jags simulation on simulated data from those, selecting in the end the best models to test on the real data.

The continuous likelihoods were

1. Double Exponential (3 models)
2. Exponential (2 models)
3. F-Dist (0 models - not suited)
4. Gamma (3 models)
5. Weibull (1 model)

Best one resulted the Exponential (5th model)

The models(extra!)

The discrete Likelihoods were:

1. Poisson (1 model)
2. Negative Binomial (1 model)

The best model resulted to be the Negative Binomial, that presented a diagnostic also better than the continuous ones.

The best continuous model (on the real data)

The Exponential model obtained the lowest DIC = 67192 also on the real data, n.eff all greater than 1000 and it was already the best on the simulated data.

The model has 8 parameters and the mcmc simulation was performed on 3 chains, with a burnin of 2000 over 39k simulation and thin factor of 10.

The best discrete model (on the real data)

The Negative Binomial obtained the lowest DIC = 79334 also on the real data, n.eff all greater than 900 and it was already the best on the simulated data.

The model has 8 parameters and the mcmc simulation was performed on 3 chains, with a burnin of 5000 over 15k simulation and thin factor of 10.

The diagnostic for the Exp Model

1. The trace plots of the chains are well mixed and in top of each other
2. The autocorrelation goes fast to 0.
3. In the geweke plot the points are always in the Z-score best interval.
4. From the gelman plot we see values below 1.1, the chains seems to converge at 5000 iterations.
5. We computed point estimates for the parameters, confidence intervals and Hypothesis testing (the leaguePoints are important features).
6. The coverage was 0.9655, it can still lead to further improvements, if we had the winrate of the players the model should improve drastically.

The diagnostic for the NegBin Model

1. The trace plots of the chains are well mixed and in top of each other
2. The autocorrelation goes fast to 0 (much better than the Exp model).
3. In the geweke plot the points are always in the Z-score best interval.
4. From the gelman plot we see values below 1.1 and almost immediate convergence(we had large burnin) .
5. The coverage was 0.64 , it can lead to further improvements, if we had the winrate of the players the model should improve drastically.

Thank you!

author

Leonardo Placidi