League of Legends Match Outcome Prediction

Lucas Lin
lucaslin@stanford.edu
Department of Computer Science, Stanford University

Abstract

We use gradient boosted trees and gradient boosted trees with logistic regression to predict the match outcomes of the popular online multiplayer game, League of Legends. Features are extracted from the data that Riot Games API exposes--including champions picked for the game, player role information, mastery levels for the players' champions, and in-game character statistics. We find that using only pre-match knowledge (champions, masteries, roles, spells) from the very start is only a weak predictor of match outcome but using in-game statistics, the model becomes a strong predictor.

Data

Riot Games provides a public API endpoint to access nearly all kinds of data that would be available to see in the official game client match history and player data from which we extracted:

- Champions played for a certain role in each match
- Spells selected in each match
- Champion mastery levels
- Stats during the game
  - Damage, Kills, Assists, Gold, etc.

Features

Pre-Match Knowledge

- Champions played per role
- Summoner spells selected
- Mastery level per champion

In-Game Knowledge

- Damage dealt per player
- Kills/Assists/Deaths per player
- Gold obtained per player
- End champion level
- More similar data

Feature Vector Template

```
{
   “[stat_name]_[side]_[LANE]_[ROLE]”: [Value]
   “assists_red_TOP_SOLO”: 10
   ...
}
```

Models

Gradient Boosted Trees (GBT)

We use Friedman’s modification to the gradient boosting method and fit a decision tree, represented by $h_m(x)$, to residuals. The update rule for the model F is show below [1].

$$F_m(x) = F_{m-1}(x) + \sum_{j=1}^{J} \gamma_{jm} h_m(x) I(x \in R_{jm}), \quad \gamma_{jm} = \arg\min_{\gamma} \sum_{x_i \in R_{jm}} L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i))$$

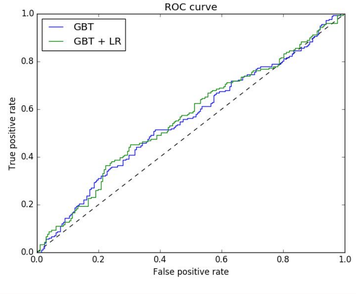
Gradient Boosted Trees with Logistic Regression (GBT+LR)

This altered model attempts to see whether or not our data can be predicted through a linear model after it has been transformed by the ensemble of trees from gradient boosting. We fit the trees on the training set and then assign each leaf an arbitrary feature index in a new feature space which are encoded using one-hot encoding. The transformed data is then run through logistic regression.

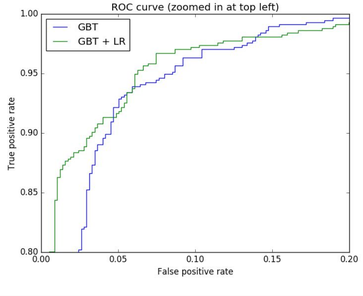
Results

Model	Train Error	Test Error
Pre-Match GBT	0.0	0.433
Pre-Match GBT+LR	0.0153	0.448
In-Game GBT	0.0	0.0608
In-Game GBT+LR	0.0	0.0642

Pre-Match Knowledge



In-Game Knowledge



Discussion

Model Reasoning

- Tree ensembles do not expect linear features and handle binary classification well because the method is using combinations of decision trees.
- Boosting makes the algorithm a good fit for handling the high dimensional space we have as well as the expected large number of training examples.

Predictor Effectiveness

- Pre-Match Knowledge
  - Overfitting due to extremely large feature vector size compared to training sample size
  - With over 100 champions and 10 different player roles in a game, representing champions in the feature vector takes 1330 distinct features
- In-Game Knowledge
  - Strong predictor of match outcome
  - ROC graph shows that the models with in-game knowledge has excellent discrimination ability

Future

In the future, it would be interesting to develop a finer sense of how accurate the prediction system becomes given the amount of data at a certain point in a match. In other words, how much more accurate would the prediction becomes if data extracted from the first *X* minutes of each match is used as part of the feature vector as compared to the first *Y* minutes as well as how the accuracy changes with respect to *X*.

References

[1] J. H. Friedman, "Greedy function approximation: a gradient boosting machine." *Annals of statistics*, 2001, pp. 1189-1232.