Starcraft Skill Assessment

Group Members: Yuting Lu 23688047, Tingyin Ding 70052496, Jiapeng Guo 94363980

1. Problem Statement

Skill estimation is basically a latent variable problem. Each player has some hidden variables that represent their skill level. Then we could combine the skills to produce a probability model of win. For this project, we estimated the skill of players in Starcraft using different graphical models considered different game features based on an empirical dataset. We evaluated our model performance based on the validation accuracy of predicted results and real results. We also evaluated the amount of games required to accurately predict the results in different aspects.

2. Referenced Resources

- Project notebook provided by the professor

  The notebook uses Stan to define the model. It assumes the skill levels follow Gaussian distribution and the winning probability follows a logistic function.

- PyStan is a python package that provides bayesian inference using Monte Carlo sampling.

- Guo, S., Sanner, S., Graepel, T., & Buntine, W. (2012). Score-Based Bayesian Skill Learning.

  The paper provided a bayesian skill estimation model that helps us to model a player's offense and defense skills separately and use these information to generate score-based match outcomes.

- Minka, Tom, Ryan Cleven, and Yordan Zaykov. "Trueskill 2: An improved bayesian skill rating system." Tech. Rep. (2018).

  Trueskill 2 is an extension of Trueskill, which incorporates additional information such as final score and player experience.

- Starcraft2 Dataset: The dataset is provided by the professor and is splitted into trains and valid. It includes the following variables: Match_date specifies the date of match happened. Match date range is from 2010 to 2017. Player_1 and player_2 refer to the nickname of players. Player1match_status and player2match_status indicate the winner and loser of the game. Player1race and player2race stand for the race of the character, which could be Zerg (Z), Protoss (P), or Terran (T). The score variable simply refers to the final score of a match.

3. Exploration

We use PyStan to build different models and learn the models through Monte-Carlo samplings. In our baseline model, we assume the players' skill levels follow a normal distribution with μ= 0 and **σ**=3. We consider the probability of one player winning over another as a Bernoulli logistics function of the difference of their skill levels. Based on our baseline models, we consider four features that potentially influence the game performance: offense/defense skills, game scores, race chosen by players (Zerg, Protoss or Terran), and the player's experience when the match happened. For each of the features, we built a new stan model and repeated the learning and evaluation process. We then evaluated all the models with the number of wrong predictions and games required for accurate prediction. For the baseline model, we also test its ability to predict a new player's skill.

3.1 Models With New Features

- When we consider **the offense/defense skill features**, we divided the player's skill into offense skill (OS) and defense skill (DS). Both skills' levels are modeled with distribution $Normal(0,3)$

  The score gained in the i-th match for playerA and playerB following the poisson distribution:

  $$ScoreA_i \sim Poisson(exp(OS_A - DS_B)), ScoreB_i \sim Poisson(exp(OS_B - DS_A))$$

  The game result is modeled as a bernoulli logistic distribution:

  $$Win_i \tilde{} BernoulliLogistic((OS_A + DS_A) - (OS_B + DS_B))$$

- As we find adding offense/defense performance well, we also consider a more complex model with score difference and overall skill estimation. Then, in addition to the above assumptions, we add $Skill_k \tilde{} Normal(OS_k + DS_k, 1)$ for player k. We model score difference with a normal distribution: $Scores_i \tilde{} Normal(Skill_A - Skill_B, 1)$

  The game result is modified as $Win_i \tilde{} BernoulliLogistic(Skill_A - Skill_B)$

- We also explored **the races** chosen by the players. There are three different races and three versions in the datasets. In each version, there are different counter relationships between races. We use two different models to examine race counter relationships. In our first model, we assigned a bernoulli distribution as the skill level for each race (Zerg (Z), Protoss (P), or Terran (T)) with each player.

$$S_p, S_t, S_z \ \sim normal(0, 3)$$

The win probability is modeled as a logit function of the skill level of the race of both players.

$$Win_{i,\, race1,\, race2} \tilde{} BernoulliLogistic(S_{race1}A - S_{race2}B)$$

In our second model, we take into account the issues that racial strengths vary in different versions. For each match happened, we simply add a weight to the factors of counter relationship (1 or 0) in our graphical model.

- We considered the player’s experience when the matches happened as well as other extrinsic factors. The players’ skill might grow over time and the most representative way to measure a player’s experience is the total matches he or she played. We considered the influence of the number of matches (M) the player experienced as an exponential function that correlates with the win probability function.

$$Win_i \tilde{} BernoulliLogistic(exp(\lambda * M_A) * Skill_A - exp(\lambda * M_B) * Skill_B)$$

3.2 Evaluations Metrics

For evaluating our model performances, we counted the loss of our models and loss vs. number of matches. For the TrueSkill based model, we further measured its ability to predict skills for new players and the amount of matches needed to accurately predict the result.

- **Evaluations of the models**

We used two different measurements given the facts that training and validation datasets have the same sets of players. In the first approach, we went through each record in the validation datasets and calculated the estimated skill level difference from the training datasets to determine who will win. We divided the right predicted match results by the total match amounts as our accuracy, which is **loss per game**. In the second approach, we consider each of the two opponent pairs and calculate the relative win rate based on the dataset. We compare the empirical win rate with the estimated skill difference as the validation accuracy. In the second approach, we calculated both soft probability discrepancy and rigid binary difference, which is **soft loss** and **binary loss**.

The soft loss is the mean square sum of the difference between true probability calculated in the validation and the probability calculated with estimated skill levels:

$$Loss = \frac{1}{n}\sum_{i=1}^{n}(p - \widehat{p})^2$$

For Binary Loss and Loss per game, we normalize the accuracy to range(0, 1) and fit for 1000 iterations. Binary Loss compares the difference between playerA and playerB

The performance of all the models could be summarized as follows (for all of the experiment bellow, we pruned both training and validation datasets by keep 15 games for each player and keep 9 game for each opponent pair):

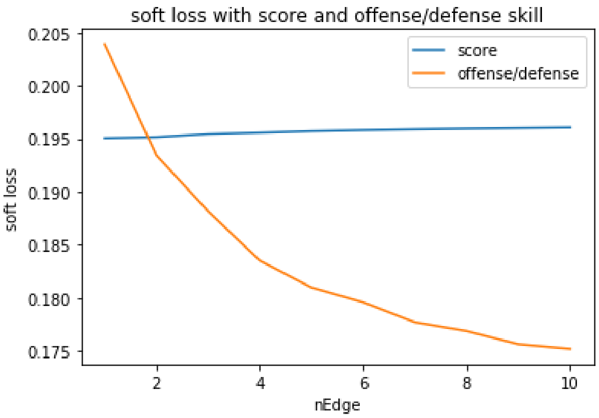
Model Features	Soft Loss	Binary Loss	Loss per game
baseline	0.1225	0.1702	0.2109
offense/defense skill	0.1296	0.1856	0.1130
With scores difference	0.3665	0.1510	0.1940
races counter relationship	N/A	N/A	0.1544 (1st model) 0.1932(2nd model)
player experience	0.2100	0.3334	0.3418

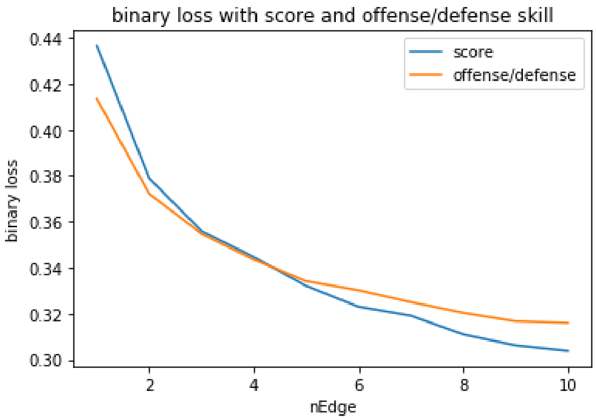
We are unable to measure the soft loss and binary loss of per opponent pairs because the races chosen by the players vary in each match. Thus we only included the loss per game for the races counter relationship feature.

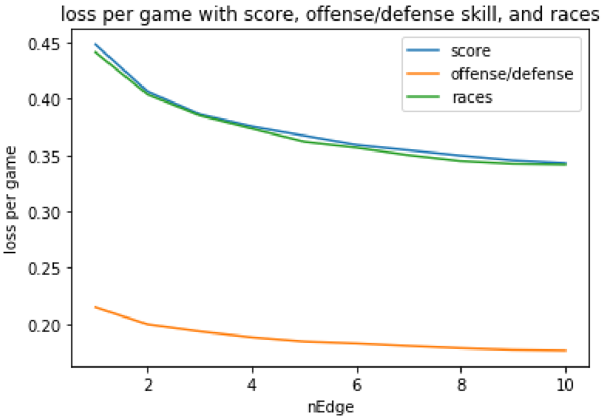
After evaluating all the models, we found that the model considered offense defense skill has the best performance. The models considered win scores and races have slightly better performance than the baseline model. The player experience model has the worst performance.

- **Games required for accurate prediction**

After we experimented with different feature models, we chose the model with best performance, which are offense/defense skill model, races model, and score model, and tried to evaluate how many games are required to accurately predict the skill level of players. We adjusted the training data by changing the amount of matches to keep for each player (nEdge) and measured the performance of different nEdges values. For accurate evaluation, we keep all the validation datasets instead of pruning, which leads to increased loss compared with the loss we calculated in the previous tables. The result graph is as follows. We did not include the soft loss and binary loss for the same reason stated above.



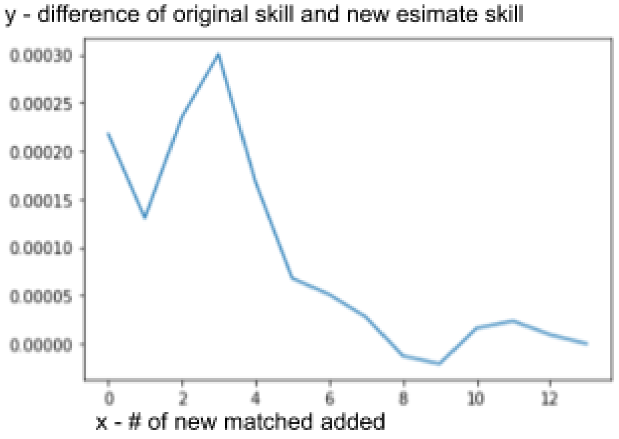




As the amount of games to keep increases, the loss of our model decreases and the accuracy increases. As the model learned from more match datas, it did a better job in correctly predicting the new match outcomes. From the graph, we can also tell that the offense/defense skill model has the best performance compared to other models.

- **Determine a new player skill**

We also try to evaluate how quickly our model can determine a new player's skill level. In our experiment, we chose one player and calculated its original skill level based on all the data. Then, we prune out the player and delete all the match data of the player. We keep the match information that was pruned out and incrementally add it to our graphical models and relearn the model. We measure the loss by the difference of the original skill level and the new estimated skill level. The experiment result is as follows.



It is obvious that as more match results were added to the model, the model did a better job at accurately predicting the skill. Usually with **5** new matches, we can determine a new player’s skill level. In this experiment, we pruned out only one player and calculated all the results. More experiments that prune out more players might further prove our result.

4. Conclusion

In our skill estimation project, we developed different graphical models to measure the influence of different features of the game. We mainly explore four features which are races, player experience level, offense/defense skill and scores for each match. Based on our experiment results, races and offense/defense skill have the most influence. We evaluate the influence of the amount of training data on model performance. We noticed that as the training data increased, the model performance might increase initially with adequate amounts of new matches but decrease when more conflict data was induced. We evaluate the amount of new matches needed to estimate the level of a new player. We found that 4 or 5 new matches is an appropriate amount to correctly predict the relative skill of a new player.