Graphical Model Final Project Report

Group Members:
Adarsh B. Shankar : 11972805
Bao Q. Tran: 30972707

Problem definition
In this project we are going to rate players' skill in chess based on tournament results and use the same model to predict the outcomes for different rounds of the tournament.

Technical approach
To do this, we are going to use a graphical model using PyGms and use the model to estimate the skill level amongst players and then use that data to predict different match results.

Dataset
For this project, we obtained a raw dataset found on github:
[DATA607/Chess_Player_Matches.csv at master · kfolsom98/DATA607 (github.com)]
This dataset consists of 5 columns player_num, round,outcome, opponent_num and opponent_pre_rating. There are 64 players with their IDs stored in the first column who compete against an opponent with their IDs stored in the 3rd column with a total of 408 rows i.e 408 games.

Performance Ratings
Performance rating is a hypothetical rating that is calculated after the games of a single tournament.
Each player's rate before the tournament begins is as follows:

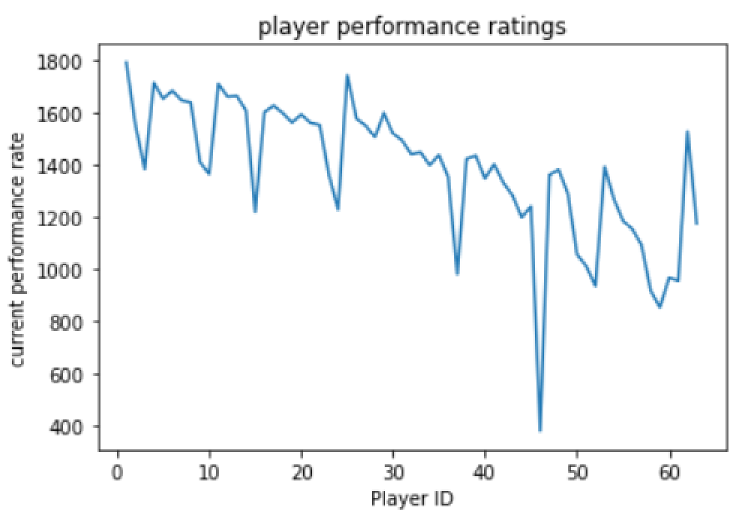

Figure 1

Many chess organizations use the “algorithm of 400” In order to recalculate a player’s performance rating after each game. For each win/loss, add your opponent's rating plus/minus 400 then divide this sum by the number of played games. After 7 games, the players’ ratings are as follows:

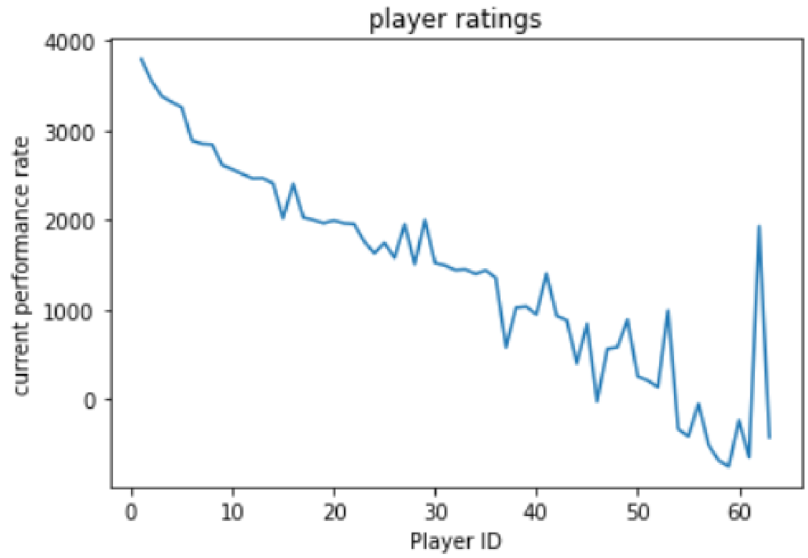

Figure 2

Mean Skill Estimations of 64 Chess Players

As shown in the plot below, player 4’s skill estimation is the highest while player 53 or 59 is expected to have the lowest skill level. This plot also clearly demonstrates that the rating for each player after 7 games is expected to look like figure 2.

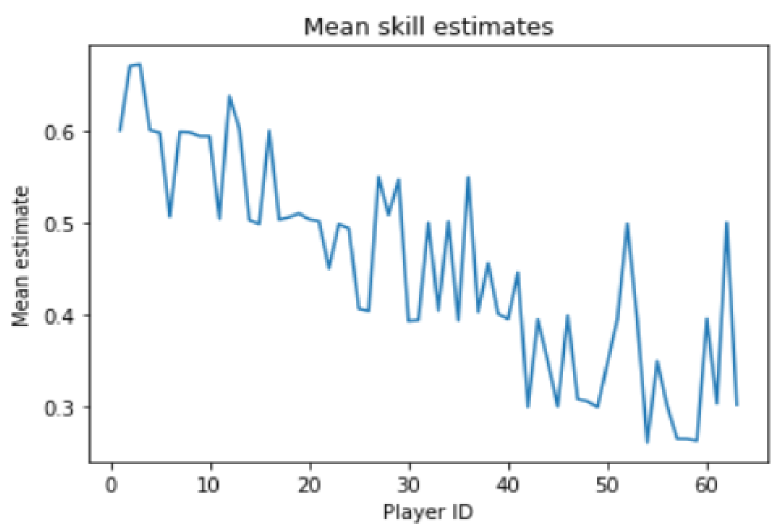

Figure 3

Calculating Bias and predicting match outcomes:
We calculated the bias by taking the difference between the two players' skill levels dividing it by 10. We tried using a different bias function based on the players' pre-match rating but these ratings from the dataset were widely inaccurate and gave worse results. We tried a variety of bias functions, 4 in total and here are the best and worst functions and the results each one gave:

```
#Based on skill estimation
def bias(x, y):
    return (x-y)/10
```

Results:
Correct: 296/408
Error Rate = 27.45%

```
#Based on opponent pre-rating skill
def bias2(x,y):
    return (x-y)/1000
```
Results:
Correct: 223/408
Error Rate = 45.34%

We can see that the bias function based on our skill estimation from the graphical model defined before gives more accurate results than the bias function based on the opponents' pre-match rating. This is because our model has adapted to the information based on the current dataset and predicts the outcome based on the training data.

Conclusion:

In conclusion, our findings are below:

- We were able to determine the estimated mean skill of 64 players who played a total of 408 games.
- Creating the bias function based on training data proved to give more accurate results for predicting the match outcome over the data pre-rating
- Based on the skill level we can predict the winner of the next game had an error rate of 0.27.
- We found that around 400 games at least would need to be added to determine a player's mean skill and predict the outcome of the match between them.