INFO 6205 Program Structure & Algorithm

ENGLISH PREMIER LEAGUE RANKING SYSTEM

GROUP MEMBERS:

001837603 Yifan Zhang

001054503 Leming Li

Table of Contents

I Introduction

- What is EPL?

It referred to as the English Premier League or the EPL outside England, is the top level of the English football league system. Contested by 20 clubs, it operates on a system of promotion and relegation with the English Football League (EFL). Seasons run from August to May with each team playing 38 matches (playing all 19 other teams both home and away).[1] Most games are played on Saturday and Sunday afternoons.

- What is Poisson Process?

A Poisson Process is a model for a series of discrete event where the average time between events is known, but the exact timing of events is random. The arrival of an event is independent of the event before (waiting time between events is memoryless).

A Poisson Process meets the following criteria (in reality many phenomena modeled as Poisson processes don't meet these exactly):

1. Events are independent of each other. The occurrence of one event does not affect the probability another event will occur.
2. The average rate (events per time period) is constant.
3. Two events cannot occur at the same time.

For example, suppose we own a website which our content delivery network (CDN) tells us goes down on average once per 60 days, but one failure doesn't affect the probability of the next. All we know is the average time between failures. This is a Poisson process that looks like:

Example Poisson Process with average time between events of 60 days.

- What is Poisson distribution?

Poisson distribution is a is a discrete probability distribution that expresses the probability of a given number of events occurring in a fixed interval of time or space if these events occur with a known constant mean rate and independently of the time since the last event[1], which means a series of discrete event where the average time between events is known, but the exact timing of events is random. A Poisson Process needs to meet these criteria:

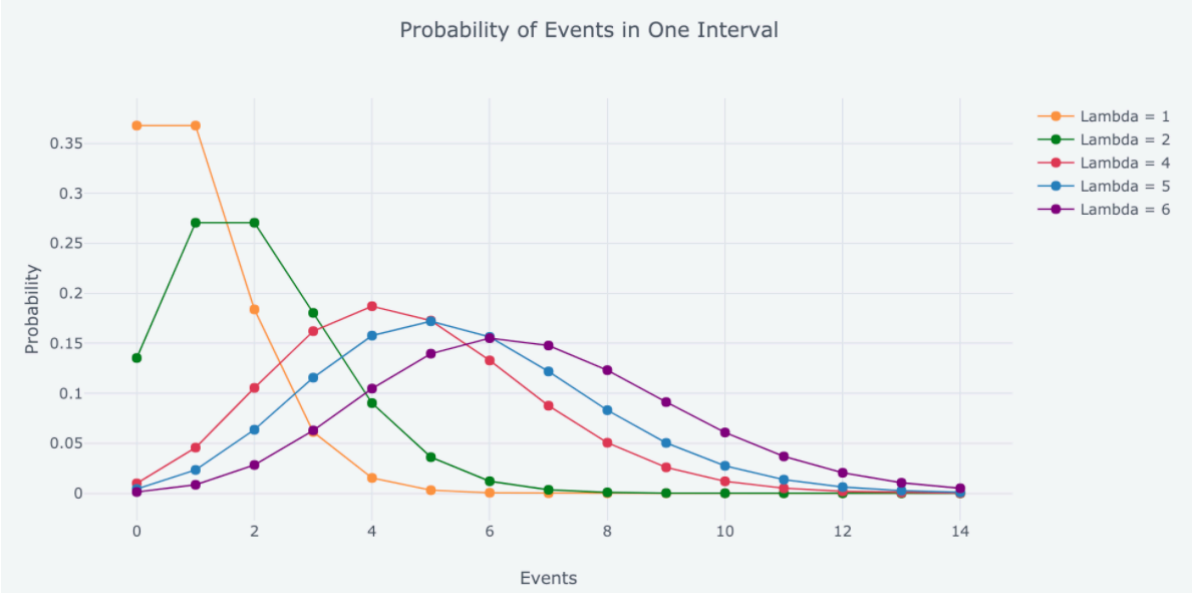
1. Events are independent of each other. The occurrence of one event does not affect the probability another event will occur
2. The average rate (events per time period) is constant
3. Two events cannot occur at the same time

The Poisson Distribution probability mass function gives the probability of observing k events in a time period given the length of the period and the average events per time:

$$P(k\ events\ in\ interval) = \frac{\lambda^k e^{-\lambda}}{k!}$$

where λ is the average number of events per interval, k = {0,1,2,3…}

The below graph is the probability mass function of the Poisson distribution showing the probability of a number of events occurring in an interval with different rate parameters.



Probability Mass function for Poisson Distribution with varying rate parameter.

The most likely number of events in the interval for each curve is the rate parameter. This makes sense because the rate parameter is the expected number of events in the interval and therefore when it's an integer, the rate parameter will be the number of events with the greatest probability.

λ : The most likely number of events in the interval for each curve is the rate parameter.

Application: We can use the Poisson Distribution mass function to find the probability of observing a number of events over an interval generated by a Poisson process.

II Problem Statement

Aim of the Project:

- Develop a ranking system which is able to evaluate the probability $P(x_i, x_j)$ where x_i would beat x_j if they met in a head to head matchup at neutral territory.
- It is desirable also that the value of P is not just a single number, but a probability density function (pdf), in other words there should perhaps be some sort of bounds on the P, maybe a uniform pdf between two values.
- Build a table of elements ordered to illustrate the win-lose relationship between each team
- Predict the residual matches of 2019-2020 season in light of COVID-19 which has abruptly terminated.
- Form a ranking table for English Premier League of 2019-2020 season.

EPL-Dataset

In the http://www.football-data.co.uk/ website we referenced below, we get the clear dataset, which is csv files includes all the seasons data from 2000 to 2020 year. All data is in csv format, ready for use within standard spreadsheet applications.

Key terms to results data:

Term	Description
Div	League Division
Date	Match Date (dd/mm/yy)
Time	Time of match kick off
HomeTeam	Home Team
AwayTeam	Away Team
FTHG and HG	Full Time Home Team Goals
FTAG and AG	Full Time Away Team Goals
FTR and Res	Full Time Result (H=Home Win, D=Draw, A=Away Win)

III Project Description

First of all, we think football match meets the critiria of Poisson distribution. Because Previous football match results can't influence the next one, which means they are independent of each other. At the meantime, the results of one match for home team and away team is symmetrical, in other words, they cannot occur at the same time.

The data what we need in our compute model is:

- For each team:
  tHG = total goals as home team
  tAG = total away goals as away team
  tHL = total goals lose as home team
  tAL = total goals lose as away tem
  tN = total number of game played;
- For all history matches data:
  allHG = total number of goals of home team,
  allAG = total number of goals of away team,
  allN = total number of matches.
- Average goals:
  AvgTeamHG = Average home goal = tHG / tN
  AvgTeamAG = Average away goal = tAG / tN
  AvgTeamHL = Average home goals lose = tHL / tN
  AvgTeamAL = Average away goals lose = tAL / tN

  AvgAllHG = Average home goals in EPL = allHG / allHG
  AvgAllAG = Average away goals in EPL = allAG / allN
  AvgAllHL = Average home goals lose in EPL = AvgAllAG
  AvgAllAL = Average away goals lose in EPL = AvgAllHG

  Because the number of goals a home team scores will equal the same number that an away team concedes, the average number of goals an average team concedes is simply the inverse of the above two numbers (AvgTHG , AvgTAG).

- Attack Strength for each home team:

$$\text{Home Attack Strength} = \frac{\text{AvgTeamHG}}{\text{AvgAllHG}}$$

- Attack Strength for each away team:

$$\text{Away Attack Strength} = \frac{\text{AvgTeamAG}}{\text{AvgAllAG}}$$

- Defense Strength for each home team:

$$\text{Home Defense Strength} = \frac{\text{AvgTeamHL}}{\text{AvgAllHL}}$$

- Defense Strength for each away team:

$$\text{Away Defense Strength} = \frac{\text{AvgTeamAL}}{\text{AvgAllAL}}$$

- Predict home goals:

$$\text{PreHG} = HomeAttackStrength * AwayDefenseStrength * AvgAllHG$$

- Predict away goals:

$$\text{PreAG} = AwayAttackStrength * HomeDefenseStrength * AvgAllAG$$

Then we use the PreHG and PreAG as $\lambda 1$ and $\lambda 2$ to put into the poisson distribution equation. K1 and k2 would be different score result. And multiply them to get the possibility of each correspond score result. Inside the possibility table, it would be a highest possibility result, which stands for the most likely goal for each team in this prediction match.

IV Implementation

Here is the project source code distribution gragh.

Basically, we a Model package which store 4 models for data storation. And LoadData class is designed for loading data from the dataset folder. Then the simulateEPL package contains 2 class to predict the residual matches in this season and the poisson distribution model relatively. The main class is our kernal file to run our system. It will call LoadData to load data first, then get all needed data using our own models designed, then put them into predictor to do predictions. And in main class, we print the rank table and complete the final match tabkle for the current season and store them into csv file in Result folder. And also take one postponed match as example to print the possibility table for this match, which is Arsenal vs Leicester.

Key Data Structures:

```
public class Predictor {

    private Map<String, Map<String, MatchData>> thisSeasonMatchData;
    private Set<MatchData> pastMatchData;
    private Map<String, TeamData> teamMap;
    private AllMatch allMatchInfo;
public class Main {
    private static LoadData loader;
    private static Predictor predictor;
    private static Map<String, Map<String, PredictMatch>> predictMatches;
    private static Map<String, TeamData> finalTeamMap;
```

Basically, we primarily use some key data structure in Main.java and Predictor.java. As we can see, they are HashMap, HashSet, ArrayList, etc. The Map<String, Map<String, PredictMatch>> is really useful to store the predict match data for a hometeam. We can easily get all match data of each other team by index a home team.

V Project Results:

The final results of our ranking system project are stored in the Result folder like below:

EPL_RankingTable

EPL_RankingTable is the final ranking table after our predictions of all the matches abandoned by COVID-19. We also print it in console like this:

2019-2020 English Premier League Table

Rank	TeamName	GP	W	D	L	F	A	GD	Score	Points
1	Liverpool	38	31	6	1	78	26	+52	97.96	99
2	Man City	38	23	8	7	82	36	+46	76.12	77
3	Chelsea	38	21	8	9	63	41	+22	66.33	71
4	Leicester	38	18	10	10	65	36	+29	64.68	64
5	Man United	38	20	10	8	56	31	+25	64.16	70
6	Arsenal	38	15	17	6	54	40	+14	59.41	62
7	Tottenham	38	13	15	10	56	47	+9	55.44	54
8	Sheffield United	38	13	11	14	33	33	0	53.46	50
9	Wolves	38	10	19	9	47	45	+2	52.17	49
10	Everton	38	11	13	14	44	54	-10	50.73	46
11	Burnley	38	14	9	15	40	48	-8	49.31	51
12	Crystal Palace	38	11	14	13	32	41	-9	49.14	47
13	Newcastle	38	10	15	13	33	50	-17	47.19	45
14	Southampton	38	12	9	17	42	60	-18	45.29	45
15	West Ham	38	7	12	19	41	60	-19	38.02	33
16	Brighton	38	6	12	20	33	49	-16	37.43	30
17	Bournemouth	38	7	13	18	36	58	-22	36.60	34
18	Watford	38	7	14	17	33	54	-21	36.57	35
19	Aston Villa	38	8	9	21	40	66	-26	36.19	33
20	Norwich	38	6	10	22	30	63	-33	30.14	28

GP: GamePlayed
W: Wins
D: Draws
L: Loses
F: GoalsFor
A: Goals Against
GD: Goal Difference
Score: The score is the score computed by our model which using possibility, so it would be more neutral and resonable;
Points: while points is the point accumulated by highestly possible result for each predition match.

finalTable.csv

finalTable.csv is the final match table which records all match data like below:

	A	B	C	D	E
1	HomeName	AwayHome	FTR	HomeGoals	AwayGoals
2	Liverpool	Brighton	H	2	1
3	Liverpool	Aston Villa	H	2	0
4	Liverpool	Sheffield Un	H	2	0
5	Liverpool	Norwich	H	4	1
6	Liverpool	West Ham	H	3	2
7	Liverpool	Newcastle	H	3	1
8	Liverpool	Leicester	H	2	1
9	Liverpool	Burnley	H	2	0
10	Liverpool	Tottenham	H	2	1
11	Liverpool	Man United	H	2	0
12	Liverpool	Bournemout	H	2	1
13	Liverpool	Crystal Palac	H	2	0
14	Liverpool	Southamptor	H	4	0
15	Liverpool	Watford	H	2	0
16	Liverpool	Wolves	H	1	0
17	Liverpool	Arsenal	H	3	1
18	Liverpool	Chelsea	D	1	1
19	Liverpool	Everton	H	5	2
20	Liverpool	Man City	H	3	1
21	Brighton	Liverpool	A	0	1
22	Brighton	Aston Villa	D	1	1
23	Brighton	Sheffield Un	A	0	1
24	Brighton	Norwich	H	2	0
25	Brighton	West Ham	D	1	1
26	Brighton	Newcastle	D	1	1
27	Brighton	Leicester	A	0	2
28	Brighton	Burnley	D	1	1
29	Brighton	Tottenham	H	3	0
30	Brighton	Man United	A	0	1
31	Brighton	Bournemout	H	2	0
32	Brighton	Crystal Palac	A	0	1
33	Brighton	Southamptor	A	0	2
34	Brighton	Watford	D	1	1
35	Brighton	Wolves	D	2	2
36	Brighton	Arsenal	A	0	1
37	Brighton	Chelsea	D	1	1
38	Brighton	Everton	H	3	2
39	Brighton	Man City	A	0	1

finalPossibilityTable

finalPossibilityTable is the table which store the possibilities of home win, draw and away win.And the mostly likely result of this match and the corresponding possibility data for all matches in this season. And if the highest possibility is 1.0, which means this match is already happened before.

	A	B	C	D	E	F	G
1	HomeName	AwayHome	Possibility of Home Win	Possibility of Draw	Possibility of Away Win	MostLikely Result	Mostlikely Result Possibility
2	Liverpool	Brighton	1	0	0	2 vs 1	1
3	Liverpool	Aston Villa	0.705558256	0.181642422	0.112779734	2 vs 0	0.127084962
4	Liverpool	Sheffield United	1	0	0	2 vs 0	1
5	Liverpool	Norwich	1	0	0	4 vs 1	1
6	Liverpool	West Ham	1	0	0	3 vs 2	1
7	Liverpool	Newcastle	1	0	0	3 vs 1	1
8	Liverpool	Leicester	1	0	0	2 vs 1	1
9	Liverpool	Burnley	0.770801504	0.153321672	0.075835489	2 vs 0	0.142762376
10	Liverpool	Tottenham	1	0	0	2 vs 1	1
11	Liverpool	Man United	1	0	0	2 vs 0	1
12	Liverpool	Bournemouth	1	0	0	2 vs 1	1
13	Liverpool	Crystal Palace	0.682856277	0.192417361	0.124713235	2 vs 0	0.125347155
14	Liverpool	Southampton	1	0	0	4 vs 0	1
15	Liverpool	Watford	1	0	0	2 vs 0	1
16	Liverpool	Wolves	1	0	0	1 vs 0	1
17	Liverpool	Arsenal	1	0	0	3 vs 1	1
18	Liverpool	Chelsea	0.42752225	0.275603223	0.296874352	1 vs 1	0.129604634
19	Liverpool	Everton	1	0	0	5 vs 2	1
20	Liverpool	Man City	1	0	0	3 vs 1	1
21	Brighton	Liverpool	0.186128923	0.231753189	0.582115195	0 vs 1	0.123773318
22	Brighton	Aston Villa	0	1	0	1 vs 1	1
23	Brighton	Sheffield United	0	0	1	0 vs 1	1
24	Brighton	Norwich	1	0	0	2 vs 0	1
25	Brighton	West Ham	0	1	0	1 vs 1	1
26	Brighton	Newcastle	0.382594887	0.273464741	0.343940217	1 vs 1	0.12952028
27	Brighton	Leicester	0	0	1	0 vs 2	1
28	Brighton	Burnley	0	1	0	1 vs 1	1
29	Brighton	Tottenham	1	0	0	3 vs 0	1
30	Brighton	Man United	0.154449914	0.217050978	0.628494235	0 vs 1	0.127263206
31	Brighton	Bournemouth	1	0	0	2 vs 0	1
32	Brighton	Crystal Palace	0	0	1	0 vs 1	1
33	Brighton	Southampton	0	0	1	0 vs 2	1
34	Brighton	Watford	0	1	0	1 vs 1	1
35	Brighton	Wolves	0	1	0	2 vs 2	1
36	Brighton	Arsenal	0.180441464	0.223141606	0.5964127	0 vs 1	0.116346299
37	Brighton	Chelsea	0	1	0	1 vs 1	1
38	Brighton	Everton	1	0	0	3 vs 2	1
39	Brighton	Man City	0.205919868	0.235797928	0.558279902	0 vs 1	0.117107221

Possibility score table of a example match

And in the main.java, we take a example match and try to output all posible score results and its corresponding possibility could be in this match, which is Arsenal vs Leicester.
In this case, we can clearly see all possible results our system computes.

```
Example to predict: Arsenal vs Leicester
-------------------------------------------------------
Home Team Goals     Away Team Goals     Result Possibility
0                   0                   0.04
0                   1                   0.04
0                   2                   0.02
0                   3                   0.01
0                   4                   0.00
0                   5                   0.00
0                   6                   0.00
0                   7                   0.00
0                   8                   0.00
0                   9                   0.00
0                   10                  0.00
1                   0                   0.09
1                   1                   0.09
1                   2                   0.04
1                   3                   0.01
1                   4                   0.00
1                   5                   0.00
1                   6                   0.00
1                   7                   0.00
1                   8                   0.00
1                   9                   0.00
1                   10                  0.00
2                   0                   0.10
2                   1                   0.10
2                   2                   0.05
2                   3                   0.01
2                   4                   0.00
2                   5                   0.00
2                   6                   0.00
2                   7                   0.00
2                   8                   0.00
2                   9                   0.00
2                   10                  0.00
3                   0                   0.08
3                   1                   0.07
3                   2                   0.04
3                   3                   0.01
3                   4                   0.00
3                   5                   0.00
3                   6                   0.00
3                   7                   0.00
3                   8                   0.00
3                   9                   0.00
3                   10                  0.00
4                   0                   0.04
4                   1                   0.04
4                   2                   0.02
4                   3                   0.01
4                   4                   0.00
4                   5                   0.00
4                   6                   0.00
4                   7                   0.00
4                   8                   0.00
4                   9                   0.00
4                   10                  0.00
5                   0                   0.02
5                   1                   0.02
5                   2                   0.01
5                   3                   0.00
5                   4                   0.00
```

VI References

1. Haight, Frank A. (1967), Handbook of the Poisson Distribution, New York, NY, USA: John Wiley & Sons, ISBN 978-0-471-33932-8
2. Probability density function: https://www.youtube.com/watch?v=ZA4JkHKZM50&feature=youtu.be
3. EPL: https://en.wikipedia.org/wiki/EPL
4. EPL-data set: http://www.football-data.co.uk/englandm.php
5. Poisson Distribution: https://en.wikipedia.org/wiki/Poisson_distribution
6. More understanding in Poisson distribution: https://towardsdatascience.com/the-poisson-distribution-and-poisson-process-explained-4e2cb17d459
7. Using Poisson Distribution to predict football match: https://www.pinnacle.com/zh-cn/betting-articles/Soccer/how-to-calculate-poisson-distribution/MD62MLXUMKMXZ6A8