Duckworth/Lewis Module

Prakhar Gupta, 17855, MTech AI

September 15, 2021

Abstract

The Duckworth–Lewis–Stern method (DLS) is a mathematical formulation designed to calculate the target score (number of runs needed to win) for the team batting second. Earlier methods developed to predict the target runs, either preferred the 2nd innings team (Average run rate method) or the first innings team (Most productive method). Duckworth-Lewis method (implemented since 1992), is better than the previous methods because of the following reasons-

- It is fair to both sides.
- Is independent of the first team's scoring pattern.
- Takes both "overs to go" and "wickets in hand" to compute the resources.
- Most importantly it is understandable by all parties involved

Duckworth-Lewis method view total runs that can be scored as a function of overs to go (u) and wickets in hand (w) as the net value of the resources, and then fit the data accordingly.
In this assignment, have created two functions, Duckworth-Lewis20Params(CSV filename) and DuckworthLewis11Params(CSV filename) to determine the parameters to model the percentage resources available to the 2^{nd} innings team. Targert score formulation is done for cases where some interruption occur in the game after it started. For each function, along with the equation parameters, have also provided the Mean square error against the ground truth for matches that took place between 1999 to 2011.

1 Findings

1.1 Data Cleaning

The data consists of 1^{st} and 2^{nd} innings data for 1423 matches from 1999 to 2011. There are 38 columns out of which only 5 following columns are useful - Match (contains matchIDs), Runs.Remaining, Innings.Total.Runs, Overs.Remaining (Calculated from overs columns), Wickets.In.Hand and Error.In.Data
Have checked for the following discrepancies in data and have removed all data for the corresponding matches.

Anomaly	# Matches Found
1st innings was completed before 45 matches	105
Final total runs from "Total.Runs" columns did not match with Innings.Total.Runs column	23
Error.In.Data flag set as 1	11
Runs.Remaining for innings 1 was more than 0, with 0 wickets in hand	2
For Innings 1, 1st over is not 1	1
For Innings 1, target score is not -1	0
For Innings 2, target score is <=0	0
Total Common Matches Removed	**130**

Figure 1: Matches with the following anomalies in their data points

From all the above cases, have removed data for 130 Matches. In addition to the above discrepancies have observed the following anomaly in data

- For 376 Matches "Total.Runs" column did not match with the total runs column computed by aggregating "runs" column.
- For 381 Matches there was a mismatch between the final runs calculated by aggregating the "runs" column to "Innings.Total.Runs" columns

For further calculation, have went along with "Innings.Total.Runs" and "Runs.Remaining" columns, which are mentioned in the data set. This is done because, the final runs mismatch from "Total.Runs" to "Innings.Total.Runs" columns was only for 23 matches, while from aggregated column was for 381 matches.

1.2 Results

For Question 1, Z0(w)[1 - exp-b(w)u] and Question 2 Z0(w)[1 - exp-Lu/Z0(w)]. Z0(W) is evaluated by taking the mean of maximum runs remaining, for all data points of 1st inning, given wickets in hand is "w". Here we have added an extra entries for the particular case of "50 overs remaining". When 50

overs are remaining, i.e., the initial case, runs remaining is equal to the total innings run.
Parameter b(w) is evaluated corresponding to all w's(wickets in hand), whereas L is independent of wickets in hand. To obtain b(w) and L, have used two methods, first one is a vanilla gradient descent method, where loss function is squared error loss, learning rate is 10^{-10} and initial values of 0.035 is taken for each case. The iterations are executed till gradient norm gets less than 0.01. Since, this method takes a lot of time, especially because of low learning rate, have used scipy optimize's, minimize function with "L-BGFS-B" method.
The parameters given in Figure 2 are used to generate the plots of Figure 3 and Figure 4.

Paramaters			
Wickets In Hand (w)	**Z0(w)**	**Question 1 b(w)**	**Question 2: L**
0	0.000	0.035	15.987
1	6.705	3.321	15.987
2	15.039	1.061	15.987
3	27.652	0.675	15.987
4	43.788	0.418	15.987
5	66.926	0.262	15.987
6	97.120	0.180	15.987
7	132.749	0.120	15.987
8	170.133	0.094	15.987
9	207.462	0.073	15.987
10	243.588	0.066	15.987

Figure 2: Parameters obtained from both the functions. Note Z(W) is same for both functions

In the below plots, Z[11] indicates the uniform decrease of resources with each over. Whereas Z[i], with i!=11, indicates the percentage of resources available with i wickets in hand.
From the above plots we can observe that

- Slope for Figure 4 around 0 overs remaining (u) for all wickets remaining cases, seems the same. This is because the slope at u=0 will be equal to L parameter for all the case. Whereas for Figure 3, the slope will be z0(w)*b(W).
- The area decreases with decrease of wickets in hand as with more wickets in hand, irrespective of the number of overs remaining, percentage resources will always be greater than less wickets in hand.
- We can also observe that for both the cases, with 1 wicket remaining the percentage resource is approximately constant and with higher wickets remaining, the resources decreases in logarithmic manner. This is expected as with more number of wickets in hand, the team has more chance of making a high score. Now with decrease of remaining overs, there will be a substantially drop in the chances of high runs, This can further be observed with the slope which increases with decrease in overs remaining.
- In Figure 5, for 2^{nd} column we can observe that slope in general decreases with increase of wickets in hand. This is because with more wickets in hand, there is always more chance of scoring good runs.
- In Figure 5, for 1^{st} column we can observe that for 2 wickets in hand, slope has decreased, which is out from the trend. This can be because with less wickets remaining, specially with 2 wickets in hand, the team is extra cautious while using their resources.

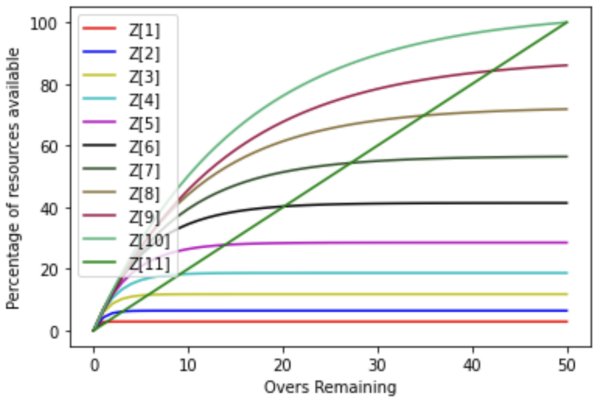

Figure 3: Percentage Available resources given parameters obtained from 1st function

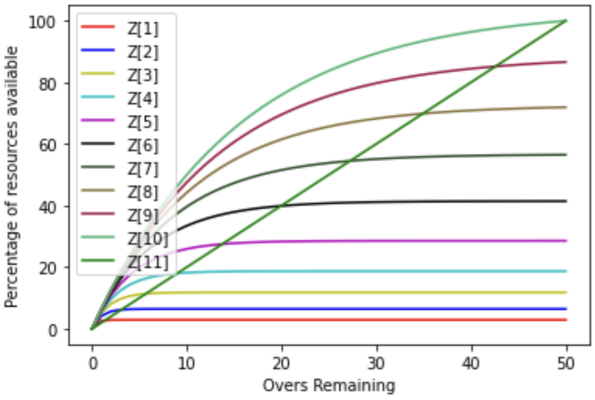

Figure 4: Percentage Available resources given parameters obtained from 2nd function

Slope with overs remaining (u)=0		
Wickets In Hand (w)	Question 1	Question 2
0	0.000	15.987
1	22.269	15.987
2	15.961	15.987
3	18.672	15.987
4	18.304	15.987
5	17.529	15.987
6	17.440	15.987
7	15.871	15.987
8	15.967	15.987
9	15.063	15.987
10	16.105	15.987

Figure 5: Slope at 0 overs remaining (u=0)

Loss error for both the questions are as follows

- **For DuckworthLewis20Params(), MSE: 1708.28**
- **For DuckworthLewis1Params(), MSE: 1719.31**

2 Code Architecture

Below have provided algorithm for first function - DuckworthLewis20Params(file:str)

- Started with some pre-processing, preprocessing()
  - Filter out 1^{st} innings data.
  - Added 50 overs remaining entries in the dataframe.
- Called func_zo_w(w) to compute Z0(w) values for every wickets in hand cases (w)
  - In case w=10, taking mean of 'Innings.Total.Runs', as all runs are remaining
  - In case w<10, after filtering out wickets in hand, took mean of maximum runs remaining of all the matches.
- For every w's, after filtering the dataframe, calling fn(b) to find the optimal b(w), taken initial value of 0.035. Used scipy library and "L-BGFS-B" method
- With Z0(w) and b(w) obtained, constructed the plot, by calling plot(y_per). y_per is a list of list which contains all the y values(Percentage resource available and the straight line).
- After constructing the plot, finding the MSE for all the error points. Obtaining predicted value from fn(b)
- Calculating the slope at u=0, which is Z0(w)*b(w)

For second function - Duckworth-Lewis11Params(file:str), the above process is same, except for the following modifications

- Along with common processing, done by pre-processing(). Some extra pre-processing is done.
  - In the dataframe have stored Z[w] values according to w's of each rows. This helps in obtaining the predicted values.
- Obtained the value parameter L from fn_ques2(L).
- Slope for each wicket in hand case at 0 overs remaining (u=0) is L.

The code can be run by importing all the libraries in terminal or other file and providing the function with the file name in string format. The file should be present in the same directory. For example the following commands can be used to run the code.
For Question 1

Z0, b
= DuckworthLewis20Params(CSV_file_name.csv in string)
For Question 2

Z0, L
= DuckworthLewis11Params(CSV_file_name.csv in string)