

Data Analytics:Assignment-1

Kishalay Das (SR Number-15938)

August 31, 2019

Duckworth-Lewis-Stern Method

Task :

Using the first innings data alone in given data set, find the best fit 'run production functions' in terms of wickets-in-hand w and overs-to-go u . Given Model is

$$Z(u, w) = Z_0(w)[1 - \exp(-Lu/Z_0(w))]$$

The loss function asked to use is sum of squared errors loss function, summed across overs and wickets. And once we fit the model, we are asked to produce a plot of the ten functions, and report the (11) parameters associated with the (10) production functions, and the total error.

Approach Used :

1. We are given data on ODI matches from 1999 to 2011 in **04_cricket_1999to2011.csv** file. First I fetched important columns from all the columns in the file. In data_processing procedure I have extracted four important features about all matches: **innings_number, remaining_runs, remaining_overs, wickets_remains**.
2. Next I take all the data points of first innings and define the loss function as sum of squared errors loss function, summed across overs and wickets. Formally we can write loss function minimization as:

$$\underset{Z_0(1), Z_0(2), \dots, Z_0(10), L}{\text{minimize}} \sum_{n=1}^N (y_n - Z(u_n, w_n, Z, L))^2$$

Where N =number of all the first inning data-points. y_n is the actual output/run and $Z(u_n, w_n, Z, L)$ is the predicted run.

3. Then I used **scipy.optimize** library to minimize the objective function defined above. I have used two different functions in this library: **scipy.optimize.least_squares** and **scipy.optimize.minimize**. In **scipy.optimize.minimize** I have further tried different methods like **L-BFGS-B, BFGS, POWEL, COBYLA, CG**. Comparison of total loss of all the methods is given as follows:

Approach Used	Total Loss
scipy.optimize.minimize(L-BFGS-B)	104818193.94097759
scipy.optimize.minimize(BFGS)	135130856.90496454
scipy.optimize.minimize(POWEL)	141407485.51578313
scipy.optimize.minimize(COBYLA)	105966517.92281651
scipy.optimize.minimize(CG)	105966517.92281651
scipy.optimize.least_squares	5671522878410885.0

Table 1: Comparison of total loss for different methods used.

Results :

From above observation I found **scipy.optimize.minimize** with **L-BFGS-B** method is giving least loss. Following is the detailed result:

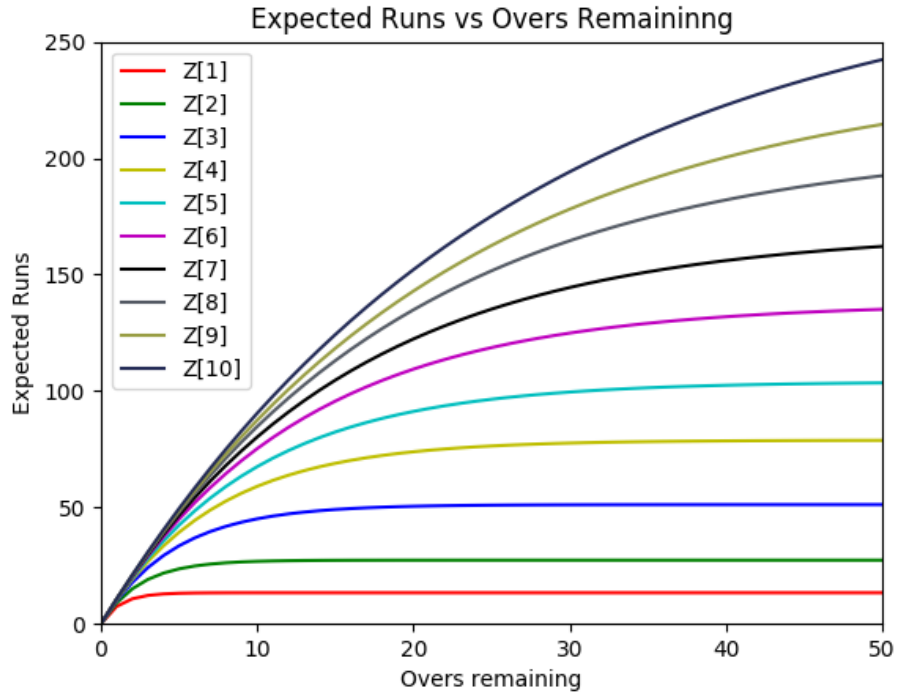
TOTAL LOSS: 104818213.15896152

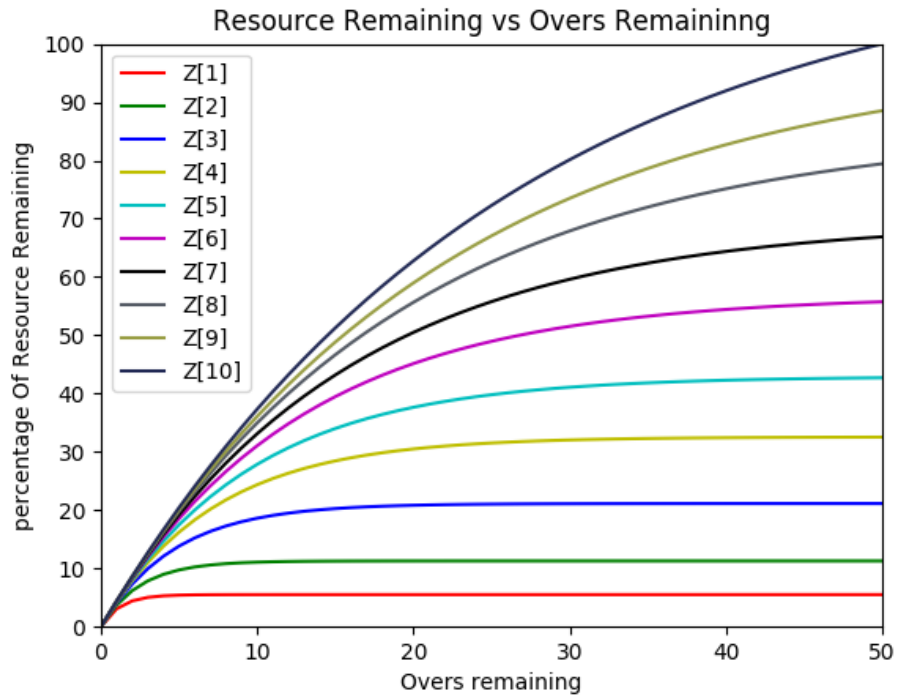
Z[1]	Z[2]	Z[3]	Z[4]	Z[5]	Z[6]	Z[7]	Z[8]	Z[9]	Z[10]	L
13.3	27.3	51.2	78.8	104.0	137.7	168.7	207.4	238.9	284.0	10.8

Table2: Optimized values of All Z and L Parameters.

Plots :

Plots for 10 functions as given below:-





github link :

Detailed plots for all the methods I used(mentioned in Table:1) and their all results are present in following github link:

<https://github.com/kdmsit/The-Duckworth-Lewis-Stern-method>