Data Analytics Assignment - 1

Achint Chaudhary, 15879

August 29, 2019

1 Problem Introduction

This Assignments aims to fitting a set of **Run Production Functions** using Duckworth-Lewis-Stern Method. Instructed to use the 1^{st} Innings data from given Data-Set of ODI matches from 1999 to 2011. We have an reasonable assumption of non-linear hypothesis as follows:

$$Z(u, v) = Z_0(w)(1 - exp(-Lu/Z_0(w)))$$

As Run production per over is independent of wickets in hand, we should have same slope for all set of functions close to Zero, for this purpose we are using shared **L** parameter between functions

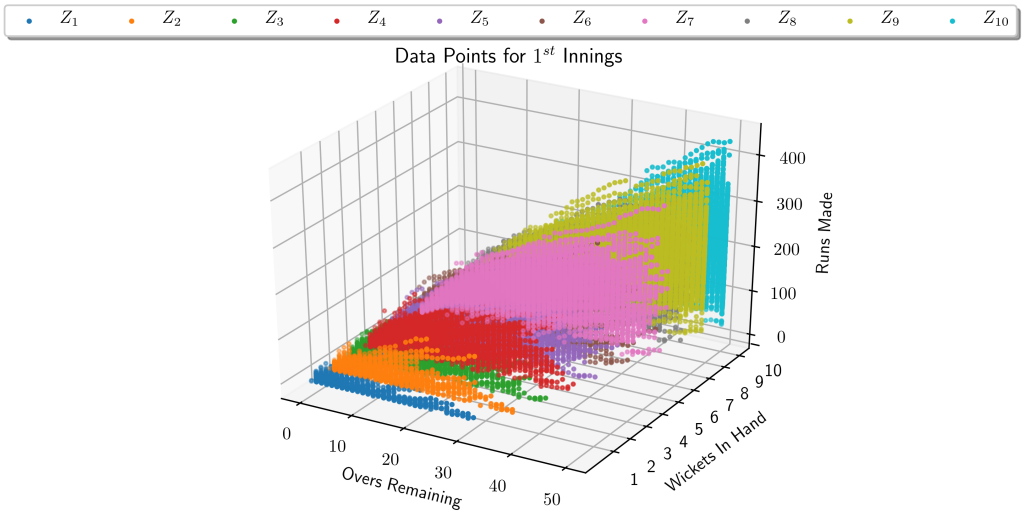
2 Preprocessing & Data-Set Plotting

Data-Set Encoding

We are using **Keras** library (not for Neural Network though!). Before we pass our training data to keras, we have encoded each data point of 2-tuple of scalars (u,w) to 2-tuple of vector and a scalar. Each scalar w is represented by its **One-Hot** representation for **Wickets in Hand**. Scalar u representing **Overs Remaining** has no changes.

Plotting and Analysis

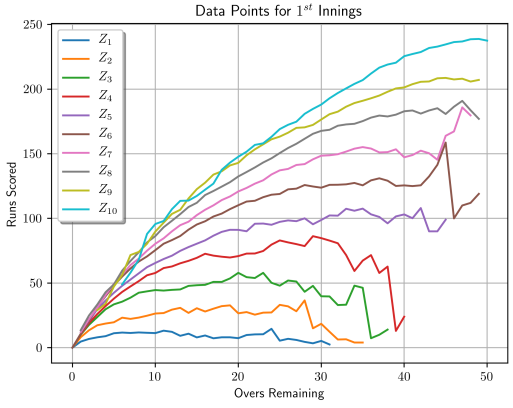
Data under consideration is plotted as different Run Production Functions for Overs Remaining. For understanding what type of approaches can be used for this non-linear regression



Above **3D** plot provides a view of all points in data under consideration. We make a note that for points where we have lower wickets in hand, there are very less or no points for some high number of overs remaining.

Next we plot mean values as a **2D** plot. Due to less or no points in some regions we have:

- No values to fit functions in some regions
- For less wickets in hand, average drops for more numbers of Overs Remaining as aggressive teams have got all out in early stages.



Note: Despite of these points not matching with our hypothesis, we have used it directly.

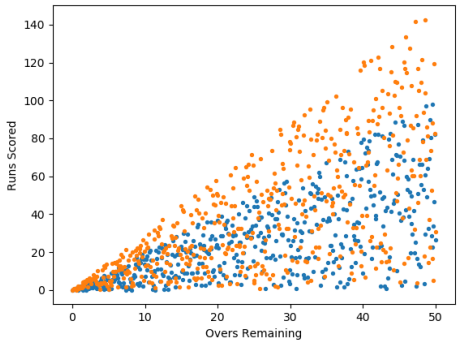
3 Experiments

3.1 Approach 1 (Mean Values)

As Wickets in Hand (w) and Overs Remaining (u) are constrained as $1 \leq w \leq 10$ and $0 \leq u \leq 50$. For each (u,v) we use mean value for (u,w) of data. This will result in **510** points in space. Such less number of points can be fitted in very short amount of time. Functions using this simplest approach can be fitted in very less amount of time

3.2 Approach 2 (All Points)

All **69k** points are taken at once and passed to keras, this approach appears to be most general but takes too long time to converge, not because due to consideration of all points but what we term as **Overlapping Triangles Problem**, which is described next.



Overlapping Triangles Problem

If we pass all points at once, standard training procedures shuffles all training data on each epoch. Shuffling is perfect for points of same wickets in hand, but for mini-batch containing data to be fit in different run production functions will change multiple components of vector Z_0. Due to this reason loss function will take too much time, even with **ReduceLROnPlateau** callback of keras. A very basic example is synthetically generated above.

3.3 Approach 3 (Sliced All Points)

Motivated from principle of considering all points at once and considering also **Overlapping Triangle Problem**. We used a sophisticated approach where points for any specific wickets in hand values are passed to keras at once, which are internally shuffles by keras. Points of other values of wickets in hand are passed later. The entire procedure is repeated several times, picking up value of w randomly from legal values of 1 to 10. This lead to faster convergence.

4 Results Analysis

In next few pages we presents values of **Sum of Squared Error** loss for different approaches we have considered. Also the learned coefficients are also presented, just to be used for future predictions.

Note: Below discussion on this page can be ignored, if we don't care about multiple local minima and fairness aspect of Duckworth-Lewis-Stern Method.

In the figures it can be observed that from moving from Approach 1 to 3, mean of predictions does shift to higher values. This is an unexpected behavior, as all three approaches converges to almost same values of SSE loss (also similar co-efficients).

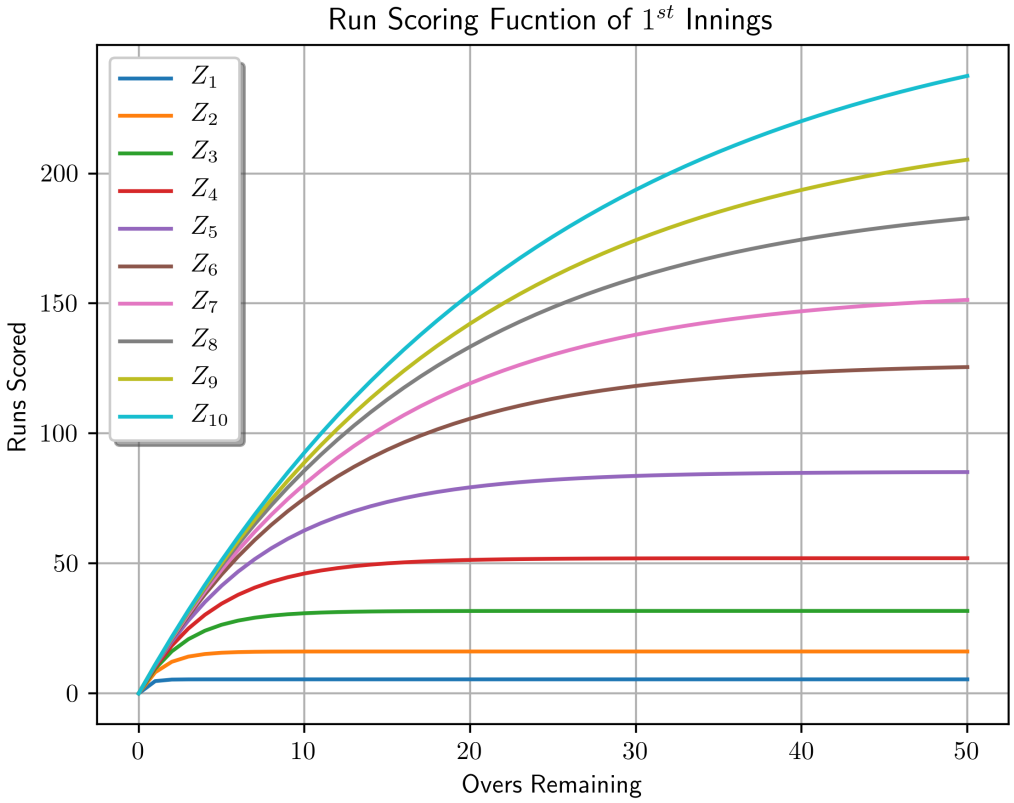
This means that we have **multiple local minima**, this comes from the fact that we have different densities of data points for different (u, w) values, this doesn't directly appear to be a problem as empirical loss is minimized. But, if we have separate validation data, we could have evaluated all three approaches separately.

To avoid this problem of converging to any arbitrary local minima, we could have change densities of points to be much equal to each other in space of (u, w). However, problems discussed due to lack of data points need to be dealt in much improved version, where the falling nature of average can also be considered.

4.1 Approach 1 (Mean Values)

L	11.334378234933611
$SSELoss$	115490008.04

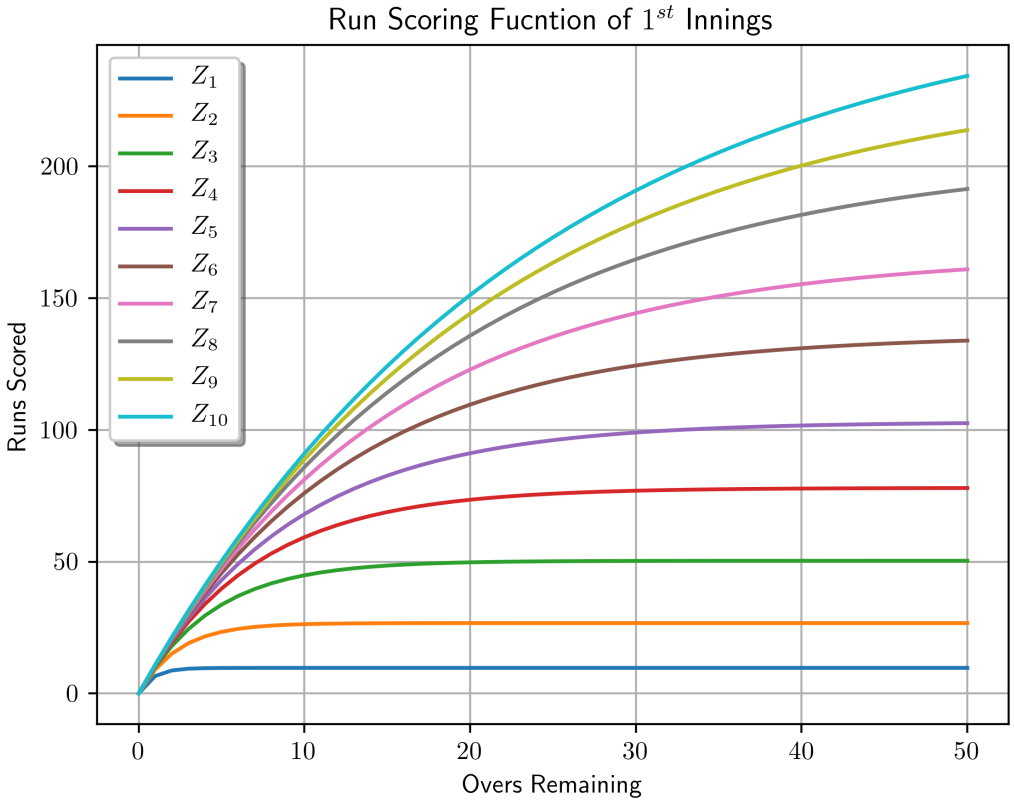
$Z_0(1)$	5.421742965288405
$Z_0(2)$	16.13945227282087
$Z_0(3)$	31.734884490367996
$Z_0(4)$	52.0106796994552
$Z_0(5)$	85.2452589702551
$Z_0(6)$	126.98456624034972
$Z_0(7)$	155.39893897534708
$Z_0(8)$	192.9581303409133
$Z_0(9)$	222.75196813256275
$Z_0(10)$	270.9662277256509



4.2 Approach 2 (All Points)

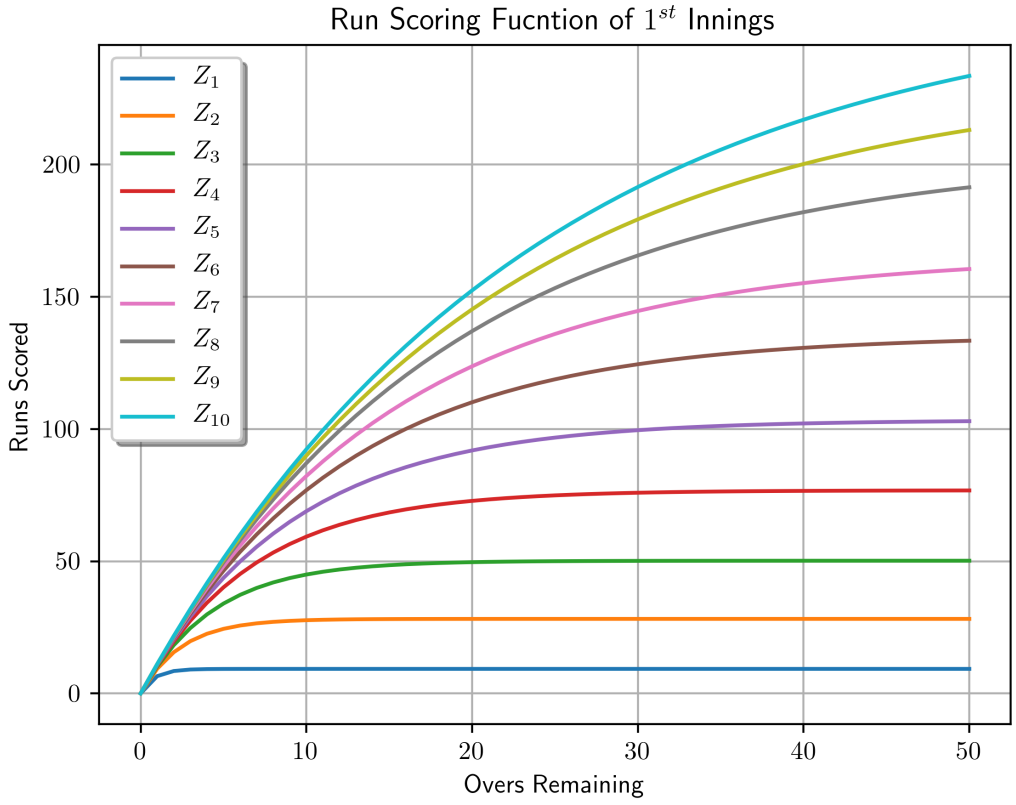
L	11.12815376553012
$SSELoss$	112016203.47

$Z_0(1)$	9.701299976483304
$Z_0(2)$	26.70647814755012
$Z_0(3)$	50.367561386413804
$Z_0(4)$	78.02370741814048
$Z_0(5)$	103.06352843350477
$Z_0(6)$	136.2132364591003
$Z_0(7)$	166.93077763648589
$Z_0(8)$	205.0599505722046
$Z_0(9)$	236.24107609783334
$Z_0(10)$	267.991747947395



4.3 Approach 3 (Sliced All Points)

L	11.356706835611366	$Z_0(1)$	9.289853680426022
		$Z_0(2)$	28.196446167131143
		$Z_0(3)$	50.187992801397684
		$Z_0(4)$	76.80387864984746
		$Z_0(5)$	103.37328676283818
$SSELoss$	112040516.34	$Z_0(6)$	135.41964545384937
		$Z_0(7)$	165.89403659438443
		$Z_0(8)$	204.00055512799528
		$Z_0(9)$	233.6059810545742
		$Z_0(10)$	264.37715039104035



Note: Despite of simplicity of **Approach 1**, it gives very similar results as compared to results of other approaches

5 Conclusion and Future Work

We have explored three approaches, and all of them have converged to almost same values of loss (also similar co-effecients).

Present **Root Mean Squared**(RMSE) loss value is around 40, which is a reasonable value, as provided only an international team is batting first with certain overs remaining and certain wickets in hand, prediction error of 40 makes sense in an ODI match.

Problems in data are explored, which can be corrected in a future work, reducing loss further, or at least providing much fairness in results due to variable densities of data points