IMPROVING DUCKWORTH-LEWIS: STATISTICAL METHODS FOR REVISING SCORE TARGETS IN LIMITED OVERS CRICKET

MATTHEW KNOWLES

1. Prelude: The Game of Cricket

Before we can discuss any maths, it is important to discuss the motivating playground for this exploration in data science: the game of cricket. Cricket takes a few forms, but the version that we look at in this project is "One Day International" (ODI) cricket. ODIs are a form of *limited overs cricket*. Meaning each team gets a set number of *overs*[1] in which to bat. We have two teams of 11 players, one team "fields" while the other bats. The batting team bat in pairs, but if one of the pair gets "out" (which can be achieved in many ways), then the next batter takes their place. This goes on until all 50 overs are completed, or the batting team run out of batters.

Sometimes however, the game is interrupted due to a adverse weather, a medical emergency or another form of interruption. If this happens when the first team is batting, known as the first innings, then the second innings is reduced to the same number of overs that the first innings had. But if this interruption occurs in the second innings, then we have a problem on our hand. Because it is unfair to expect the second team to achieve the same as the first team in less time. This is a problem that English statisticians Frank Duckworth and Tony Lewis decided to tackle. Their paper [3], outlined a method which became adopted by cricket's governing body, the International Cricket Council (ICC) in 1999. Known as the Duckworth-Lewis (D/L) method.

Duckworth and Lewis later retired, and the method was updated by Professor Steven Stern in his 2016 paper [7]. The method needed updating to account for a new form of limited overs cricket: the T20 format. In T20, each team has an innings lasting 20 overs, rather than 50. The D/L method was not designed with such a format in mind, and so the updates by Stern aimed to adress this. The method is known today as the Duckworth-Lewis Stern method, or "DLS" for short, and is what the ICC now use for resetting score targets.

2. Introduction and Aims

The cricket community has had problems with the DLS for many years, especially in T20 cricket, due to the massively increased scoring rates that the method wasn't initially designed to deal with. Nowadays, the use of data in sport in general, and in particular, cricket, has become commonplace. Teams use data in varying ways to try and give themselves an advantage going into games. The aim of this project is to use match data and train pattern recognition models on it, in order to see if using more sophisticated methods lead to more accurate score prediction in ODI cricket.

The pattern recognition methods chosen are neural networks, and clustering. Neural Networks were chosen to their reputation for predictions. They are being employed all over modern society, from predicting stock prices [6], to diagnosing cancer [5]. This flexibility and power makes neural networks a great choice of method for predicting cricket scores. The second method is clustering. The idea behind clustering is that simmilar games will appear close together when clustered based on a couple of factors. We can then look at how a game is behaving, and which cluster it is likely to end up in. Clustering is often used by online shopping websites,

Date: Autumn Term 2021.

[1]An over consits of 6 individual balls, bowled by the same person from one end of the pitch.

for reccomending customers products based on purchase history [1]. The motivation for using it is similar, except the history comes in the form of a metric in a cricket game, rather than what products may have been purchased by someone online.

The problems with DLS itself will be discussed in slightly more detail later on, but another reason it would be better to find a pattern recognition method for resetting score targets is that DLS is cumbersome. It produces a big table of values that must be input into an equation depending on the scenario a game is in. If the Nueral Network method is found to be more accurate, we just have to feed the current match state into a computer program and we're done. No need for messing around with resource tables and ratios.

3. Data

3.1. **Data Origin and Structure.** The primary source of data for carrying out this project was downloaded from "cricsheet"[2] and stored locally on a private server. In total there are 2167 individual matches of data, each in JSON format. These cover matches ranging from the 3^{rd} of January, 2004. Up to the 20^{th} of July, 2021.

Each JSON file contains a considerable amount of metadata surrounding the match in question. Along with ball-by-ball data for the entire match. We have access to attributes such as the date, where the match was played, the entire teamsheet for both teams, who the officials were, who won- and by what margin, who won the toss; and many others.

We also have the ball-by-ball data. So for every ball bowled, it gives who were the striking and non-striking batsmen, how many runs were scored and how. It also details if a wicket was taken that ball, and how.

3.2. **Pre-Processing.** In order to get data in to a usable form, Python scripts were written to read the JSON files and extract features necessary for this project. These Python scripts output CSV files that can be fed into R for easier statistical analysis. For example, when training the neural network model, a Python script was built to extract the runs scored in each over for over 1400 games. The labelled CSV data was then used by an R script to output a matrix of runrates and corresponding final runs, which was used to train the Neural Network.

4. Duckworth-Lewis

We begin by looking at the main equation that is the backbone of the Duckworth-Lewis method.

$$Z(u, w) = Z_0(w)[1 - exp(-b(w)u)] \tag{1}$$

In equation 1, u represents the number of overs the runs are scored in, w is the number of wickets lost, and b(w) is an exponential decay constant. Due to commercial reasons, the authors were unble to publish some of the mathematical constants that they used. The fact this model follows an exponential decay pattern is the key thing. Infact, this is consistent with how runs per wicket evolves.

The exponential model we see here gives rise to the first issue. The values for the constants were derived from ODI matches only, which is why this method isn't immediately applicable to T20 cricket. But it does capture the idea that as a team looses out on batters and overs, the amount of runs they can realistically score begins to dwindle. In [2], the authors find that a Bayesian approach to this method does a better job of capturing this idea of exponential decay.

[2]https://cricsheet.org/

5. Improving DLS - Statistical Pattern Recognition

The main aim of this project, as previously mentioned, is to investigate ways of producing more realistic score targets for interruption. The main way we look to achieve this is through statistical pattern recognition methods. Below we discuss each of the methods that will be investigated in the main dissertation.

5.1. **Neural Networks.** Neural networks are mathematical objects that aim to immitate how neurons work in the brain. We have an input layer of nodes, several hidden layers of nodes, and and output layer. In our model, the output layer consists of a single node. This will be a score based on the input nodes. Each node holds a value, and connects to the nodes in the proceeding layer by a connection characterised by a weight and a bias. This connection is analagous to a synapse.

We store the value of these weights as a matrix, and the biases as a column vector. Let $W^{(1)}$ be the matrix containing the weights of connections from the input layer to the first hidden layer. Further, let $\mathbf{a}^0$ be the column vector of values in the input layer, and $\mathbf{b}^1$ the biases between the input layer and first hidden layer. Then the values of the nodes in the second layer are given by the matrix equation 2.

$$A^{(1)} = W^{(1)}\mathbf{a}^0 + \mathbf{b}^1 \tag{2}$$

The same process is then repeated for the other layers. The actual *learning* of the network comes from an algorithm called "backpropogation". The idea is to minimise an error function that gets its values by comparing the network's output for a sample set of data to the actual value for that data. By changing the weights and biases to minimise this function, we obtain a set of weights and biases that give accurate predictions for future datasets.

The only issue to be wary of at this point, is that we don't actually have that much data. These models are often trained on millions of data points, but in our case we are naturally limited by the number of ODI matches that actually get played. In total we have 1436 data-points to use in training, which is not ideal, but it should still work with enough itterations of backpropogation.

Backpropogation is the algorithm used for training neural networks. The idea is to tell the network how badly the guess it has made is by using an error function,

$$E(X, \theta) = \frac{1}{2N}\sum_{i=1}^{N}(y_i' - y_i)^2. \tag{3}$$

Where θ denotes a parameter incorperating the weights and biases, and X is the input vector. The error function makes a comparrison of the predicted value y', and the actual valye for the X input vector, y. The aim of backpropogation is to find the parameter θ which minimises this function, and as a result giving the set of weights and biases that provide the best prodictions for an unseen dataset X'.

The "learning rule" for updating the weights linking node j in one layer to node i in the previous is given by 4.

$$\Delta w_{ji} = \alpha x_i \delta_j. \tag{4}$$

Where δ_j is the sensitivity for a node. Based on the sum of the weights and activations of the connections going into it. The formal definition is outlined in chapter 6 of [4]. In 4, α is the "learning rate".

5.2. **Clustering.** The next method we plan to look at is that of clustering. We can cluster games based on certain metrics in order to prpedict how as a score will evolve. The idea is to cluster our samples using the k-means clustering algorithm [4] (chapter 10.4.3). This clusters the results based on means of the sample data. The k refers to the number of clusters we break

the data into. This process works as in the following algorithm.

1: Initialise $n, c, \mu_1, \ldots, \mu_c$
2: **repeat**
3: Classify n samples according to nearest μ_i
4: Recalculate μ_i
5: **until** No change in μ_i

At present, the Nueral Network has been built, and with some tinkering should be producing results shortly. However clustering has not yet been attempted, which is why details on the metrics that plan to be used are still a little vague. Some prelimiary tests have been performed using runrate data, as with the NN model, however it wasn't particularly conclusive. For this reason, when attempting the clustering, it will be necessary to include many more metrics, such as wickets fallen, boundaries scored etc.

Identifying the correct number of clusters to use, or which metrics are the most important will be a challenge, however there are methods in Exploratory Data Analysis to help with this, such as Principal Component Analysis.

6. Discussion

After discussing with performance analysysts at several county cricket clubs in the UK, most of the data used in cricket matches for informing performance decision tends to be qualitative rather than qualitative. The field is looking to explore using deeper mathematical modelling techniques, but the issue comes when presenting the results of these to cricket players, who are naturally not familiar with higher level maths, which can make communicating the results and what that means for the team difficult.

The main result of the report will come from the comparrison between the two pattern recognition techniques outlined above, and how the method which performs better fairs against the current DLS method being used by the International Cricket Council today.

References

[1] Palaksha Anitha and Malini M Patil. Rfm model for customer purchase behavior using k-means algorithm. *Journal of King Saud University-Computer and Information Sciences*, 2019.

[2] Indrabati Bhattacharya, Rahul Ghosal, and Sujit Ghosh. A statistical exploration of duckworth-lewis method using bayesian inference. *arXiv preprint arXiv:1810.00908*, 2018.

[3] Frank C Duckworth and Anthony J Lewis. A fair method for resetting the target in interrupted one-day cricket matches. *Journal of the Operational Research Society*, 49(3):220–227, 1998.

[4] Richard O Duda. *Pattern classification / Richard O. Duda, Peter E. Hart, David G. Stork.* Wiley, New York ; Chichester, 2nd ed. edition, 2001.

[5] N Ganesan, K Venkatesh, MA Rama, and A Malathi Palani. Application of neural networks in diagnosing cancer disease using demographic data. *International Journal of Computer Applications*, 1(26):76–85, 2010.

[6] Mahdi Pakdaman Naeini, Hamidreza Taremian, and Homa Baradaran Hashemi. Stock market value prediction using neural networks. In *2010 international conference on computer information systems and industrial management applications (CISIM)*, pages 132–136. IEEE, 2010.

[7] Steven E Stern. The duckworth-lewis-stern method: extending the duckworth-lewis methodology to deal with modern scoring rates. *Journal of the Operational Research Society*, 67(12):1469–1480, 2016.