Iso-probability Criterion for Cricket Interruptions

Group Name: CrickLovers (3)

Atharv Suryawanshi Maredukonda Jagath Chandra Arjun Sagar NV
Ankush Ahirwar Adithya Ponnuraj

1 Introduction

In the game of cricket, matches are occasionally interrupted due to rain or bad lighting, leading to reduced overs of play. This affects the chances of winning for both teams. In order to tackle this problem, various methods were introduced, such as Average Run Rate (ARR) or Most Productive Overs (MPO), which often results in leaving one side at a disadvantage. The DL method was introduced in 1997, which calculated the expected number of runs in the lost overs to estimate a new target. This method was significantly better than its predecessors, yet it still had limitations. In this project, we look into a new approach to tackle cricket interruptions in a more 'fair' manner called the Iso-probability criterion. It was developed by Michael Carter and Graeme Guthrie. As the name suggests, it works on the principle of conserving the probability of winning for each team before and after the interruption. Using data from matches between 2001 and 2024, we test this method and analyze its effectiveness.

2 Methods

In order to implement this method, we define a cumulative distribution function $F(r; n, w)$ that models the number of runs r a team is expected to score, given n overs remaining and w wickets in hand. This function will be the foundation for calculating the probability of winning based on various game states, both before and after an interruption.

2.1 Defining Probability of Winning

If Team 2 has a target t and has already scored s runs, then the probability that Team 2 will win the match with the remaining n overs and w wickets is given by:

$$1 - F(t - s; n, w) \tag{1}$$

Now, suppose an interruption occurs at this point, reducing the remaining overs to n' and revising the target to t'. The iso-probability condition requires that the probability of winning remains consistent before and after the interruption. Therefore, we set:

$$1 - F(t' - s; n', w) = 1 - F(t - s; n, w)$$

This equation ensures that the probability of winning for Team 2 remains unchanged, thereby maintaining fairness.

2.2 Modeling the Run-Scoring Process

To accurately estimate $F(r; n, w)$ we model the run-scoring process with three possible outcomes for each ball, denoted by b, the number of deliveries remaining, and w, wickets in hand. We assume that the following events can occur on each ball:

- No Ball/Wide: With probability p_x, the score increases by 1 run, but b and w remain the same.
- Wicket: With probability $p(b, w)$, a wicket falls, reducing w by 1 and b by 1.
- Run Scored: With probability $q(i; b, w)$, r increases by i, while b decreases by 1 and w remains the same.

Using this approach, we build a probability model for each possible score trajectory, leading to a comprehensive cumulative distribution function for run outcomes over the course of the remaining deliveries.

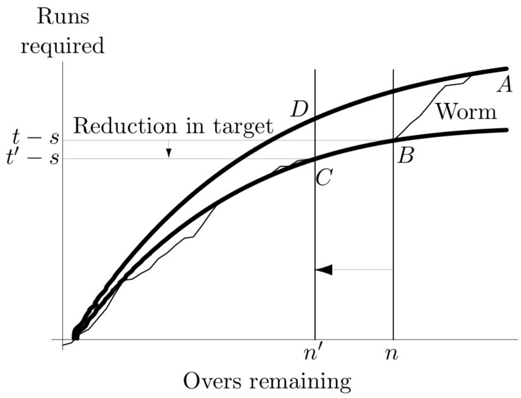

Figure 1:

2.3 Boundary Conditions

To finalize the model, we set boundary conditions that represent winning or losing situations:

- $F(r, 0, w) = 1$ if $r \geq 0$: The team loses if there are no balls remaining.
- $F(r, b, 0) = 1$ if $r \geq 0$: The team loses if there are no wickets left.
- $F(r, n, w) = 0$ if $r < 0$: The team wins if they have scored the required runs.

2.4 Estimating Probability parameters

2.4.1 Wide or No Ball

The probability of wide or no ball is estimated using the following

$$P_x = \frac{\text{Nos extra deliveries}}{\text{Nos extra deliveries} + \text{Nos legitimate deliveries}}$$

2.4.2 Wickets

The probability of loosing a wicket $p(b, w)$ while having b deliveries and w in hand is estimated using a probit model. Let the unobserved variable be $y^*_{b,w}$ defined as

$$y^*_{b,w} = \alpha_0 + \alpha_1 b + \alpha_2 w + \theta_{b,w}$$

where $\alpha_0, \alpha_1, \alpha_2$ are constants and $\theta_{b,w} \sim N(0, 1)$. According to the property of probit model, a wicket falls if $y^*_{b,w} < 0$ which occurs with a probability of $p(b, w) = \Phi(-\alpha_0 - \alpha_1 b - \alpha_2 w)$ where Φ is cumulative distribution function of a normal distribution. Using our knowledge of the game we can say: 1. $\alpha_1 > 0$ as it is more likely for a wicket to fall as the game progresses. 2. $\alpha_2 > 0$ as lower batting order more easily to fall. In order to estimate these parameters, we use a random variable y_b which is 1 if wicket falls and 0 if it does. Assuming outcomes are independent of deliveries, we get the likelihood function as follows:

$$\text{Likelihood} = \prod_{n=1} \Phi(-\alpha_0 - \alpha_1 b - \alpha_2 w)^{y_b} (1 - \Phi(-\alpha_0 - \alpha_1 b - \alpha_2 w))^{1-y_b}$$

We choose value of $\alpha_0, \alpha_1, \alpha_2$ which maximizes the value of log-likelihood for an inning.

2.4.3 Runs

The probability of scoring runs i for $i\epsilon\{1,2,3,4,5,6\}$ is given by $q(i;b,w)$ for b deliveries and w wickets in hand. Here we will use an ordered probit model with an unobserved variable $i^*_{b,w}$ as

$$i^*_{b,w} = \beta_0 + \beta_1 b + \beta_2 w + \epsilon_{b,w}$$

where the β's are constants and $\epsilon_{b,w} \sim N(0,1)$. The thresholds for runs are denoted by $\mu_0 \leq \mu_1 \leq \mu_2 \leq \mu_3 \leq \mu_4 \leq \mu_5$. The runs scored by batting team are i for $i = \{1,2,3,4,5,6\}$ if $\mu_{i-1} < i^*_{b,w} < \mu_i$, zero runs if $i^*_{b,w} < \mu_0$ and six runs if $i^*_{b,w} > \mu_5$.

$$q(i;b,w) = \begin{cases} \Phi(\mu_0 - \beta_0 - \beta_1 b - \beta_2 w), & \text{if } i = 0, \\ 1 - \Phi(\mu_5 - \beta_0 - \beta_1 b - \beta_2 w), & \text{if } i = 6, \\ \Phi(\mu_r - \beta_0 - \beta_1 b - \beta_2 w) & \\ -\Phi(\mu_{r-1} - \beta_0 - \beta_1 b - \beta_2 w), & \text{otherwise.} \end{cases}$$

Using our knowledge of the game we can say:

1. $\beta_1 < 0$ as scoring accelerates as game moves ahead
2. $\beta_2 > 0$ scoring decreases in lower batting order

In order to calculate the probability, we do the same as previous estimation and calculate the parameters which results in the maximum value of Likelihood.

2.5 Function F

Using these parameters, we construct our cumulative distribution function $F(r;b,w)$

$$\begin{aligned} F(t-s;b,w) = p_x F(t-s-1;b,w) \\ + (1-p_x)p(\alpha)F(t-s;b-1,w-1) \\ + (1-p_x)(1-p(\alpha))\sum_{i=0}^{6} q(i;\alpha)F(t-s-i;b-1,w) \end{aligned} \tag{2}$$

2.6 Code Breakdown

1. **Data Processing:**
   (a) `get_iso_data:` Preprocesses the match data, including calculating balls remaining, wickets remaining, and other match statistics.
   (b) `get_prob_of_wide_or_no_ball:` Computes the probability of a wide or no ball based on the match data.
2. **Parameter Estimation (Probit Models):**
   (a) `compute_llf:` Calculates the log-likelihood for different run values using a probit model.
3. **RunsOProbit Class:** Contains methods to fit the model to the data and predict the probability of scoring a specific number of runs given the match state.
   (a) `fit`: Fits the model to the data.
   (b) `predict`: Predicts the probability of scoring a specific number of runs based on the match state.
4. **Iso-probability Calculation:**
   (a) `construct_F:` Builds the cumulative distribution function $F(r;b,w)$ using the parameters of the run and wicket models. It uses a recursive approach to fill in the values for each combination of runs, balls, and wickets.
5. **Training the Model:**
   (a) `training:` Imports data, sets up models, and fits them to the data. It includes the training time and fitting for the wicket model, followed by the run model.

3 Results

After training the model on data of cricket matches from 2001-2024. We get the following values for the parameters:

- compute table5 table4 and figure4 from carter paper Now lets look at an example for how this

method works and compares to DL method of resetting targets. On 20th July 2003, in a cricket match Cambridge vs Oxford, Cambridge scored 190 from 50 overs and Oxford competed with a score of 162/1 from 31 overs when the match was interrupted. They needed 29 from 12 overs, a very simple target indeed. But when rain stopped with 12 overs still remaining, DL method declared Oxford as winner. The probability of Oxford winning was very high, but not 1. Alternatively our method calculates - —–prob and revised target.

In order emphasize this method, lets consider 3 grounds A,B,C with similar weather and lighting conditions. Team 1A and 1B score 250 off 50 overs and team 1C scores 180 off 50 overs. Now consider team 2A scores 120/3 in 20 overs and 2B, 2C scores 50/3 in 20 overs. Here, we will compare and contrast DL vs Iso-probability method on our trained model.

4 Conclusion

In conclusion, we have implemented an adjustment mechanism for interrupted cricket matches that preserves the likelihood of winning at the point of interruption, providing a more equitable and incentive-free alternative to existing methods. Unlike traditional systems like Duckworth-Lewis, which focus on maintaining resources (runs and wickets), this approach stresses the chance of winning as the most important metric to maintain throughout interruptions. By evaluating historical data, we validated this rule and established that it effectively maintains match integrity and player incentives, as evidenced in multiple previous matches when our strategy was used retrospectively. In certain occasions, this modification resulted in different outcomes, demonstrating the rule's potential impact on match results. In contrast to our probability-based approach, proponents of

the Duckworth/Lewis (D/L) method believe that it promotes fairness by keeping the margin of advantage throughout pauses, even if it occasionally favors the leading side. D/L advocates argue that their methodology is tactically neutral, encouraging players to stick to their customary strategy while avoiding inconsistencies caused by probability-based changes. While we accept these arguments, we have demonstrated that our strategy effectively maintains equity by balancing win probabilities. Our Iso-probability strategy provides an alternative that closely aligns the match outcome with pre-interrupted conditions, making it an equally viable option for fair target reset.

References

[1] Carter, M., and G. Guthrie. *Cricket Interruptus: Fairness and Incentive in Limited Overs Cricket Matches.*

[2] Chiappori, P. A., S. Levitt, and T. Groseclose. *Testing mixed strategy equilibria when players are heterogeneous: The case of penalty kicks in soccer.* Mimeo, University of Chicago, 2000.

[3] Clark, Stephen R. *Dynamic programming in one-day cricket — Optimal scoring rates.* Journal of the Operational Research Society, 39(3):331-337, 1988.

[4] Duckworth, F.C., and A. J. Lewis. *A fair method for resetting the target in interrupted one-day cricket matches.* Journal of the Operational Research Society, 49(2):220-227, 1998.

[5] Duckworth, Frank, and Tony Lewis. *Your comprehensive guide to the Duckworth/Lewis method for target resetting in one-day cricket.* University of the West of England, Bristol, 1999.

[6] Romer, David. *It's fourth down and what does the Bellman equation say? A dynamic programming analysis of football strategy.* NBER Working Paper 9024, 2002.

[7] *Comment on Carter M and Guthrie G (2004). Cricket Interruptus: Fairness and Incentive in Limited Overs Cricket Matches* Authored by F. C. Duckworth and A. J. Lewis. The Journal of the Operational Research Society, 56(11):1333-1337, 2005.

[8] Walker, Mark, and John Wooders. *Minimax play at Wimbledon.* Mimeo, University of Arizona, 1999.

Cricsheet Downloads https://cricsheet.org/downloads/ Dataset source