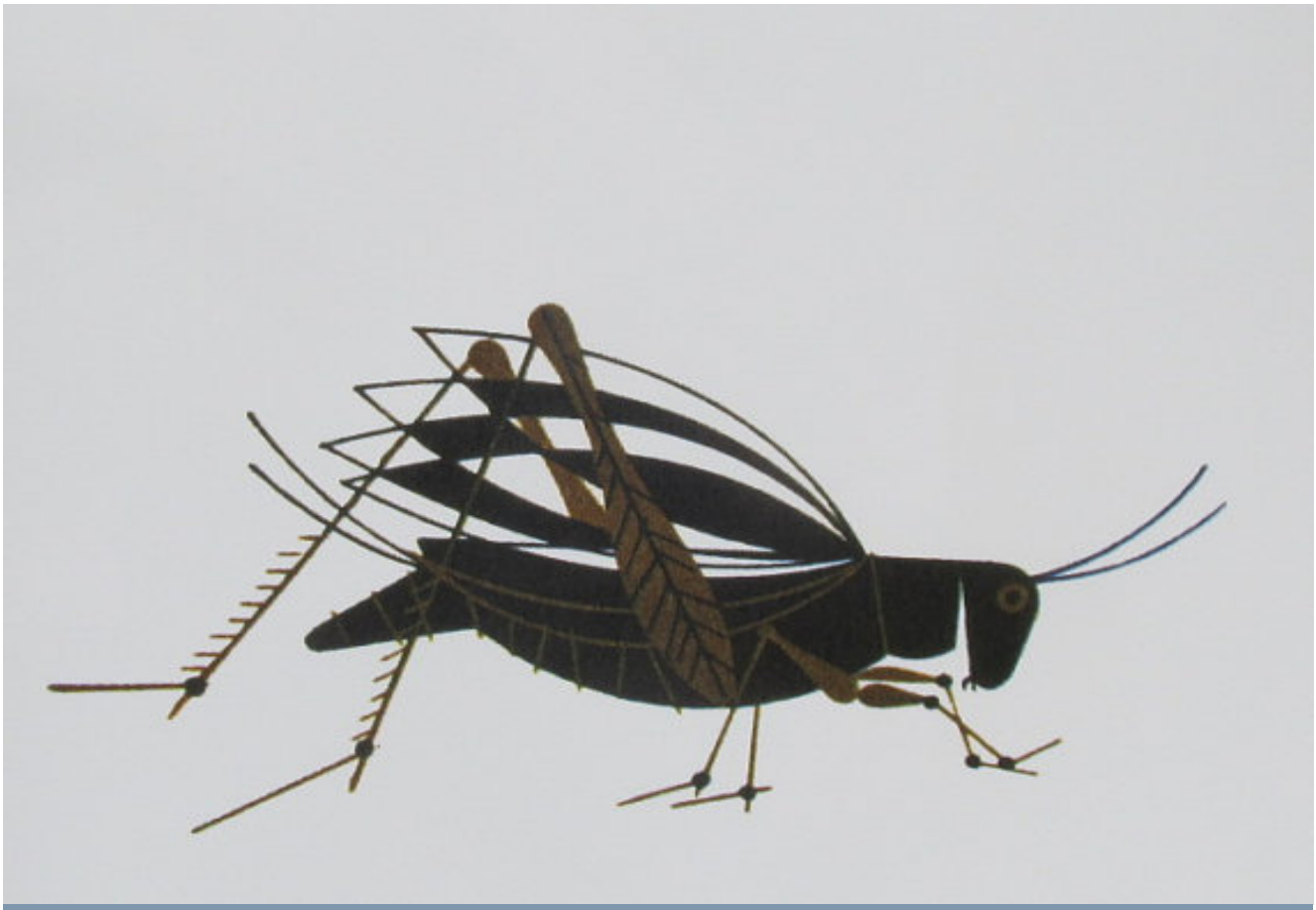


AN ANALYSIS OF CRICKET

BIG DATA | AUG – DEC 2016

SNo	Name	Usn	Class
1	Prafful U Minajigi	1PI13CS108	7B
2	S J Rajath Krishna	1PI13CS129	7B
3	Sarthak Shetty	1PI13CS137	7B
4	Shiva Deviah	1PI13CS147	7B

1. Introduction

PROBLEM STATEMENT

The aim of this project is to build a practical application, which leverages big data tools such as Hadoop, Spark, etc. to analyze and simulate IPL matches.

STRATEGIC HIGHLIGHTS

We have three different implementations for this application:

1. Naïve Approach
2. Custom Clustering approach (1 dimensional)
3. K-means Clustering approach (n dimensional, n=2, 3)

All these approaches will be explained in detail in the design section of this document.

OPERATING HIGHLIGHTS

The main tools used to carry out analysis are

- Apache Spark in Standalone mode, and
- PySpark – A Python API to communicate with Apache Spark (simply because python is our favorite language)

FUTURE ENHANCEMENTS

Testing and training the system using machine learning techniques to pick the best number of dimensions to represent a person. The more features a person is represented by, the more accuracy the system will have.

As of now, the cluster size has been fixed at K=10. An improvement over this would be to automatically select the value of K that best segregates players in the pool.

Devise a way to implement context (limited) into the simulations. For example, the simulator should be sensitive to different phases of the innings, such as the start, or the death. Probabilities will be varied accordingly.

2. Related work

In addition to the document on the class project shared on Google Drive, there were a couple of useful sources of information that went a long way towards understanding how to implement this application.

Apache Spark is an open-source clustering cluster computing framework. In contrast to Hadoop's two-stage disk-based MapReduce paradigm, Spark's in-memory primitives (RDD) provide performance up to 100 times faster.

Another major motivation to use spark was its implementation of a columnar database, Spark SQL, which uses Hive internally. While Hive on spark is undoubtedly faster for large datasets, there was no significant difference in performance for relatively small sized datasets and integration with Spark SQL meant lesser and cleaner code.

3. Design

SYSTEM BLUEPRINT

1. Data gathering – Scraped all required data from cricinfo as csv files
2. Preprocessing – Loaded data in Spark SQL
3. Simulating - Implemented three ways to do this

The approach for seen data remained the same for all three implementations. Cumulative Probability distributions were generated for each Batsman Bowler pair and stored in a pickled RDD, henceforth refered to as 'pvp'.

1. NAÏVE APPROACH

Following the approach mentioned in the google drive document, we used the pvp for seen data and returned a uniform distribution for unseen data during simulation.

2. CUSTOM CLUSTERING (1D)

We clustered Bowlers and Batmen based on their Strikerates and Economy, respectively, into 10 clusters each.

Approach 1

On encountering unseen data, we replace the batsman (for that ball) with another batsman belonging to the same cluster with the least Euclidean distance and continue to replace players until a valid pair representing a record in the pvp is found.

Approach 2

Went through each combination of clusters, and each combination of players within clusters to cumulate all events for seen data to represent a cluster. Computed the Cumulative Probability distributions for each pair of clusters ("cvc"). On encountering unseen data, find the cluster that the pair belong to and refer the record in the cvc for the respective clusters.

Extension

Implementing dynamic distribution of probability mass in cumulative probability distributions for each record in pvp where the probabilities of events keep changing as the

game progresses. We introduced a hyper parameter with which the event probability is discounted and assigned to the 'out' event. What this means in simple terms is, the longer a batsman stays on the pitch, the more likely he is to get out.

3. K-MEANS CLUSTERING (K=10; N-D, N=2, 3)

Since this is an unsupervised approach, the challenge was picking the right features to represent each player.

Clustering Batsman

Three features were picked to represent batsmen. Batsmen were represented as (Strike rate, # of 6s, # of 1s). We wanted to represent batsmen with two features, which are in correlation (based on intuition) and one without correlation and found this approach to give us fairly accurate results.

Clustering Bowlers

Two features were picked to represent bowlers. Bowlers were represented as (Economy, # of wickets). The two features picked were not in correlation, which gives more depth to each player, for analysis.

On encountering unseen data, we find the cluster to which the batsman belongs to using the Kmeans model that was trained, find all batsmen in that cluster and replace him with the most similar batsman.

Extension

Implementing dynamic distribution of probability mass in Cumulative probability distributions for each record in pvp where the probabilities of events keep changing as the game progresses. We introduced a hyper parameter with which the event probability is discounted and assigned to the 'out' event. What this means in simple terms is, the longer a batsman stays on the pitch, the more likely he is to get out.

4. Experimental Results

Here are some stats of how our application performed for an actual match.
(Scores change each time, this is just one of the results.)

RCB vs SRH, IPL Finals 2016

Actual score: SRH 208/7, RCB - 200/7

Naïve implementation – 138/3, 113/2

Custom clustering – 198/6, 191/8

K means clustering – 212/7, 197/6

[Scroll to the last page for added swag.]

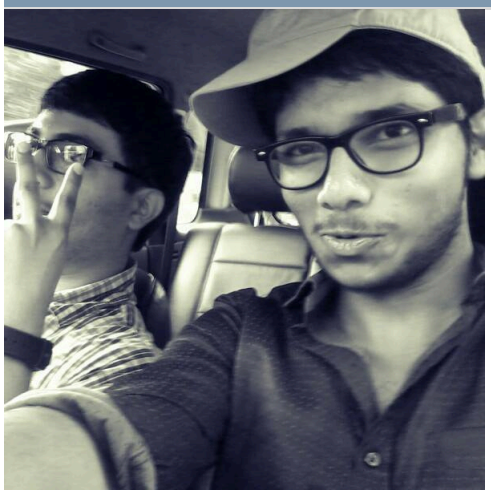
5. Evaluations

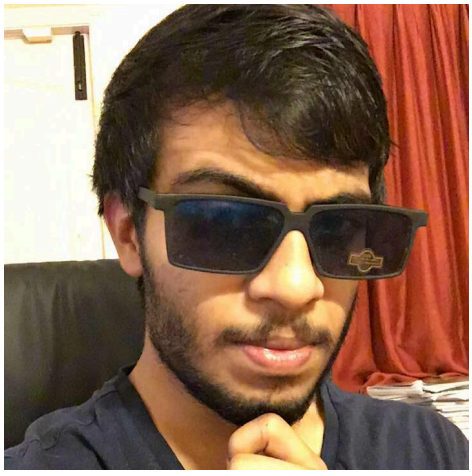
Date	Evaluator	Comments	Score

6. Checklist

SNo	Item	Status
1	Source code documented	
2	Source code uploaded to CCBD server	
3	Recorded video of demo	
4	Instructions for building and running the code. Your code must be usable out of the box.	
5	Dataset used for project uploaded. Please include a description of the dataset format. This includes input file format	

Appendix: Added Swag

PRAFFUL UM 1PI13CS108	S J RAJATH KRISHNA 1PI13CS129
	

SARTHAK SHETTY 1PI13CS137	SHIVA DEVIAH 1PI13CS147
	

Thanks!