

UE15CS314: Big Data Class Project

(IPL ANALYSIS)

Done by :

Rahul Pillai 01FB15ECS224

Parashara R 01FB15ECS202

Omkar D 01FB15ECS198

Part 1(getting the data):

* To get the ball by ball data we went to cricsheet.org website and downloaded .yaml files for the past 10 IPL seasons.
* We wrote python scripts to parse this data and get player vs player info in the following format

**“batsman,bowler,0,1,2,3,4,5,6,balls,runs,strikerate,dismissals”**

and we put this into a csv file and stored this in hdfs.

* Then to get the player info we considered only the players of the 2016 season and web scraped all their information from cricbuzz using a mozilla plugin called “imacros” and we got a csv file of all the current player’s information in this format.(current meaning only the players of the 2016 season (around 90 players)).

“**playername,team,bat,M,Inn,NO,Runs,HS,Avg,BF,SR,100,200,50,4s,6s**” (for batsman related information)

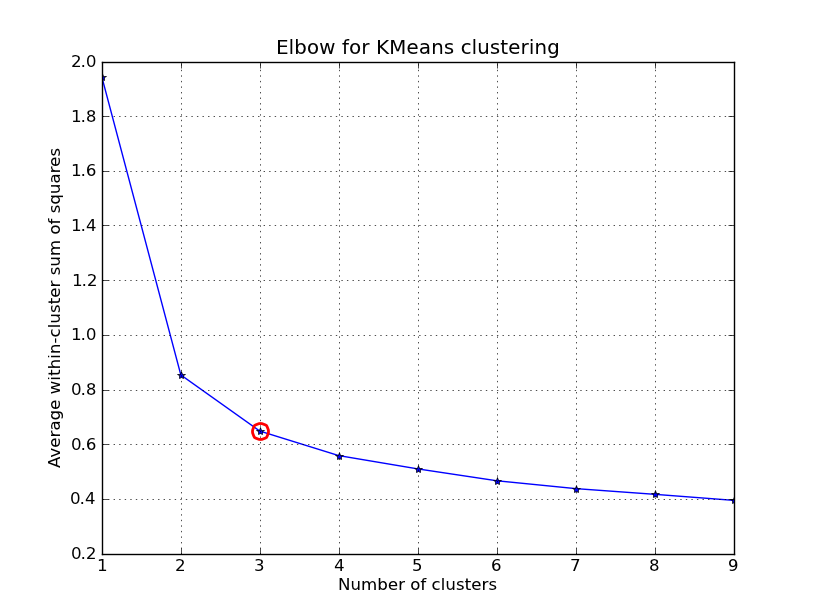
“**playername,team,bowl,M,Inn,B,Runs,Wkts,BBIruns,BBIwckts,BBMruns,BBMwckts,Econ,Avg,SR,5W,10W** “(for bowler related information)

* We uploaded both the player information and the player vs player information into hdfs at the end of this stage

Part 2(clustering into batsman and bowler groups):

* Initially we had decided to have 12 batsman and 12 bowler clusters for the k-means clustering ,but then we found that there is an algorithm called “elbow method” with which we can find out the optimum no of clusters for the k-means clustering.
* From the “elbow graph” we found out that having 5 batsman clusters and 5 bowler clusters

(a sample graph)



* We tried using a number of different technologies to get the cluster information.
* For clustering the player information into groups we considered all the features available in the player information csv file ie

“**M,Inn,NO,Runs,HS,Avg,BF,SR,100,200,50,4s,6s**” (for batsman related cluster information)

“**M,Inn,B,Runs,Wkts,BBIruns,BBIwckts,BBMruns,BBMwckts,Econ,Avg,SR,5W,10W** “(for bowler related cluster information)

* We tried using the following technologies:
  + Python
  + Spark
  + R
  + pyspark
* Eventually after applying the “k-means” algorithm we dumped the cluster data into 4 csv files. With the following formats.

**Cluster centroid information for batsman**:

**#cluster,M,Inn,NO,Runs,HS,Avg,BF,SR,100,200,50,4s,6s**

**Cluster centroid information for bowler:**

**#cluster,M,Inn,B,Runs,Wkts,BBIruns,BBIwckts,BBMruns,BBMwckts,Econ,Avg,SR,5W,10W**

**The new player information for batsman with cluster no:**

**#cluster,playername,team,bat,M,Inn,NO,Runs,HS,Avg,BF,SR,100,200,50,4s,6s** (just added the cluster number he belongs to here)

**The new plyaer information for bowler with clusterno:**

**#cluster,playername,team,bowl,M,Inn,B,Runs,Wkts,BBIruns,BBIwckts,BBMruns,BBMwckts,Econ,Avg,SR,5W,10**(just added the cluster number he belongs to here)

* These 4 csv files were then uploaded into the hive for later retrieval

Part 3(getting the probabilities):

* After retrieving these csv files from hive we tried to calculate all the necessary probabilities of runs being scored and wickets taken.
* So our first step was modifying the existing player vs player csv file to also include the batting cluster no and the bowling cluster no for each row’s batsman and bowler.
* To estimate the probabilities we just used the maximum likelihood estimator for all the existing pairs of batsman and bowlers from the player vs player info ie.

P(event) = (no. of times event occurred)/(total number of events)

Eg P(1 run being scored)=(no of times 1 run was scored by that batsman)/(no of balls faced against that bowler).

* So for each pair of batsman and bowler we were able to calculate probability of 1 run scored………,probability of 6 runs scored ,probability of a wicket falling.
* But then the problem was “what if there is another new batsman bowler pair who have never faced each other before”
* For this we needed the cluster information from the previous step and our approach was to take all the existing batsman-bowler pairs we have already seen before (i.e. the player vs player information) and for each possible batsman-bowler cluster combination figure out the new “cluster vs cluster” information which can be used in the case we ever see new pairs we haven’t encountered yet.
* To get this “cluster vs cluster” information for each of the 25 possibilities (ie 5 batsman clusters and 5 bowler clusters) we considered the following policy:
  + For each one of the 25 cluster, cluster combinations say a batsman from the ith cluster facing a bowler from the jth bowling cluster.
  + We consider all the rows from player vs player information which already fall in this category of the batsman from the ith cluster facing a bowler from the jth cluster and take all their existing player vs player records and the take the average of all of it
* This way for each one of the 25 cluster vs cluster combinations we now have a “predicted” player vs player information.
* The next step we did was find all the possible legal batsman, bowler combinations and get a set difference with all the existing batsman, bowler combinations so that we get all the possible “new” pairs of batsman and bowler we haven’t seen before which could happen as we simulate the match.
* Then we mapped each new combinations with their cluster numbers and found out which cluster vs cluster player information each pair had to take.
* Then we found the probabilities of each event i.e, P(0 run)……P(6 run),P(wicket falling).for all these “new” combinations and dumped all the data into a new csv file which contains all the probabilities for all possible batsman ,bowler pairs we can ever encounter.
* Just to simplify this problem we considered only the players of 2016 IPL season.
* Final format of this file containing all the probabilities is as follows:

**batsman,bowler,p0,p1,p2,p3,p4,p5,p6,pdismissals,balls**

Part 4(simulating the with all the existing data ):

* Now that we have all the probabilities of each possible batsman ,bowler pair in a file we can start to simulate the match.
* The main part of the simulation is the policies we use to predict 2 things:
  + “how to figure out how many runs are scored off each ball given which bowler is bowling to which batsman”
  + “how to figure out at what stage can the wicket fall given the batsman on strike and the current bowler bowling”
* **Policy to predict runs:**
  + for each batsman, bowler pair we have row of probabilities ie (p0,p1,p2......p6)

so we first generate a random probability number between 0 and 1 say x. we then bucketize all the probabilities using a cummulative probability and then see in which bucket does x fall in . that is the run being scored eg

* + kohli vs dalesteyn
  + prob cummulativeprob
  + p0=0.1 0.1
  + p1=0.2 0.3
  + p2=0.15 0.45
  + p3=0.15 0.6
  + p4=0.1 0.7
  + p5=0.2 0.9
  + p6=0.1 1.0
  + so if the random number x=0.5 ==> a 3 has been scored!
* **Policy to predict the fall of a wicket:**
  + for each batsman ,bowler pairs we only need the P(dissmissals) for this.
  + firstly we need to access the batting average of each batsman say BAi and store it somewhere and the use this to compute the (1-threshold value) for each batsman BTi.(Batsman threshold)
  + (where BTi=(BAi/BAmax)-0.01) so for a good batsman like Kohli has a high batting average so his (1-threshold) value will be close to 1 therefore his threshold will be close to 0 but not exactly 0.
  + so for good batsman like him that would mean that he takes a very long time to get out as it would take many balls for his survival probability to go below the threshold value.
  + For every bowler a batsman BAi faces we keep multiplying the probability of him being not out vs that bowler (which is just 1-P(dismissals) for that pair) with the existing “survival probability”
  + The moment the “survival probability” of that batsman Bai goes below his threshold BTi at that moment we say a wicket has fallen.
  + So as we simulate we give higher priority to a wicket falling as opposed to a run being scored
  + i.e we predict both the runs scored and the survival probability for each ball but the moment we know that the survival probability<threshold value we consider that a wicket has fallen and ignore the runs prediction the batsman has made.

EXTRA FEATURES ADDED TO THE SIMULATION(JUST FOR FUN😊):

* Added an interactive ball to ball commentary for various scenarios in a random way.
* Added a prematch ceremony where a random host talks about the pitch conditions.
* Added a match summary where we get to see the entire summary of the entire simulation from who won the game , who took how many wickets, how did the wickets fall etc.