Project Report
on
Game Prediction - Cricket

Harshit Shah

Computer Science Department
Rochester Institute of Technology
Rochester, NY 14623

Ajinkya Rode

Computer Science Department
Rochester Institute of Technology
Rochester, NY 14623

1 Abstract

Our project is about predictive analysis for a win/lose condition of a particular team in the game of Cricket using Decision Tree Induction algorithm. Based on the historical data, we predict whether the team will win or lose the near games against the specific teams. The goal is to predict a result for a particular team which competes with other seven different teams. In our project, we are using Weka as a classification tool and J48 decision tree algorithm, which is equivalent to C4.5 algorithm. After building a model in Weka, we are using that model to test specific instances for the prediction using User Interface made in Java Web.

Keywords: *Decision Tree, C4.5, J48, Cricket*

2 Introduction

The sports word is known for the vast amounts of statistics that are collected for players, teams and matches. There are also many types of statistics a Cricket player will have data for runs scored, matches played, average, strike rate etc. for each match. Such data are collected almost all sports for analysis and knowledge creation for continuous improvement. Hence, sports are ideal for our project. We have chosen Cricket for the purpose of research project.

[1] Factors such as winning the toss and the home team advantage affecting the results of ODI games have been studied. In our project, Weka's J48 algorithm is applied to historical data for purposes of model-fitting. Our data consists of a few set of the ODIs played between nations for the time period starting January 1995 to the end of the 2010 matches. Some of the matches were deleted from the analysis due to certain reasons such as abundance of bad weather or when the one team was much superior to the other (ranked teams playing non-ranked teams). Tied games were also deleted from the analysis. Due to the continuous update of cricket rules, we chose, in particular, to use this most recent data. We also added weather as one of the factors to predict the result of the game. Finally, we tested our prediction model using UI developed in Java based on Spring MVC framework.

3 C4.5 Algorithm background and literature review

Original applications of decision trees were in domains with nominal valued or categorical data but today they scattered in multiple domains with numeric, symbolic, and mixed-type attributes. The widespread popularity of C4.5 in data mining makes us to use it in our project. So for Exam 2, our focus is to thoroughly examine C4.5 algorithm.

3.1 C4.5 Introduction

C4.5 [4] is a suite of algorithms for classification problems in machine learning and data mining. It uses historical data for classification. C4.5 is targeted at supervised learning: Given an historical dataset where instances are described by collections of attributes and belong to one of a set of mutually exclusive classes. C4.5 learns a mapping from attribute values to classes that can be applied to classify new, unseen instances. All of the data constitutes "training data," so that we can learn mapping and apply it on other, new instances with values of only attributes to predict the

value for the class random variable.

C4.5, designed by J. Ross Quinlan, is so named because it is developed from the ID3 approach [5]. A decision tree is a series of questions systematically arranged so that each question queries an attribute and branches based on the value of the attribute. At the leaves of the tree are placed predictions of the class variable. Apart from inducing trees, C4.5 can also state its trees in comprehensible rule form. Further, the rule postpruning operations supported by C4.5.

3.2 Description

C4.5 is a suite of algorithms which contains C4.5, C4.5-no-pruning, and C4.5-rules-with many features. Consider a set S of different cases consisting of different values. The C4.5 will generate the tree using initial data from set S using divide and conquer algorithm to split or classify the values depending on the case as below:

- If majority of the cases belong to the same class or value of S is minimum
  then the leaf of the tree is labeled with the most frequent class in S.
- Else choose a test, based on a single attribute with multiple outcomes.
  → Make the test result as the root with two branches for each outcome of the test
  → Then apply the above method recursively for each subset.

Usually multiple methods are used to perform the second step. C4.5 uses two heuristic criteria to rank possible tests: information gain, which minimizes the total entropy of the subsets $\{S_i\}$ (but is heavily biased towards tests with numerous outcomes), and the default gain ratio that divides information gain by the information provided by the test outcomes.

3.3 Algorithm

The above algorithm shows the working of C4.5 where it shows how the entire set is partitioned into subsets. The root node is the classifier using which the subsets are formed by tree partitioning. This procedure continues recursively until the subsets are pure and fall in one class leading to which the growth of the tree is terminated.

3.4 Inducing tree from the data

Let us look at the various choices involved in inducing such trees from the data.

Algorithm 1.1 C4.5(D)

Input: an attribute-valued dataset D

1: Tree = $\{\}$
2: **if** D is "pure" OR other stopping criteria met **then**
3: terminate
4: **end if**
5: **for all** attribute $a \in D$ **do**
6: Compute information-theoretic criteria if we split on a
7: **end for**
8: a_{best} = Best attribute according to above computed criteria
9: Tree = Create a decision node that tests a_{best} in the root
10: D_v = Induced sub-datasets from D based on a_{best}
11: **for all** D_v **do**
12: Tree_v = C4.5(D_v)
13: Attach Tree_v to the corresponding branch of Tree
14: **end for**
15: **return** Tree

Figure 1: Algorithm Description

- **What types of tests are possible?** C4.5 is not restricted only to binary tests, and allows tests with two or more outcomes. If the attribute is Boolean, the test induces two branches, while the attribute is categorical, the test is multivalued, but these multiple values can be grouped into a smaller set with one class predicted for each option. If the attribute is numerical, then the tests are again binary-valued.
- **How are tests chosen?** Information-theoretic criteria such as gain and gain ratio are used to determine the best test greedily. Gain means reduction in entropy of the class distribution due to applying a test and gain ratio is a way to correct for the tendency of gain to favor tests with many outcomes. The default criterion is gain ratio. At each point in the tree-growing, the test with the best criteria is greedily chosen.
- **How are test thresholds chosen?** As stated earlier, for Boolean and categorical attributes, the test values are simply the different possible instantiations of that attribute. For numerical attributes, the threshold is obtained by sorting on that attribute and choosing the split between successive values that maximize the criteria above. Not all splits between successive values need to

be considered, it only needs to consider splits between successive different values after they are sorted.

- **How is tree-growing terminated?** There are two ways. When the all instances that covered by a specific branch are pure or the number of instances fall below a certain threshold.

- **How are class labels assigned to the leaves?** The majority class of the instances assigned to the leaf is taken to be the class prediction of that subbranch of the tree.

3.5 Information Gain

The attribute with the highest informational gain is computed using the following formulas:

Entropy:

$$E(S) = \sum_{i=1}^{n} -Pr(C_i) * log_2 Pr(C_i)$$

where,
E(S) ← information entropy of S
n ← number of classes in S
$Pr(C_i)$ ← frequency of class C_i in S

Gain:

$$G(S, A) = E(S) - \sum_{i=1}^{m} Pr(A_i) E(S_{Ai})$$

where,
G(S,A) ← gain of S after split on attribute A
m ← number of values of attribute A in S
$Pr(A_i)$ ← frequency of cases that have A_i value in S
$E(S_{Ai})$ ← subset of S with items that have A_i value

3.6 C4.5 Features

The C4.5 algorithm improves the ID3 algorithm by allowing numerical attributes, permitting missing values and performing tree pruning.

- **Missing values:** Missing values deals with handling incomplete data or data in the form '?'. Imputation is a term used with respect to missing values for estimating the important feature that is missing using available data. Based on this imputation, distributed imputation is a type, in which we split our data into multiple instances with a different value for the missing feature.

- **Tree Pruning:** Tree pruning is used to get a small and consistent tree with specific features. It helps in correct classification as the tree after pruning contains only the specific features. The features that lead to inconsistent tree classification are eliminated either before of after constructing a tree. There are two types of pruning as follows:

  - **Pre-pruning:** Stops tree construction as it encounters irrelevant attributes.
  - **Post-pruning:** Eliminates branches after tree construction, that leads to irrelevant attributes.

3.7 Advantages and Disadvantages of C4.5

- **Advantages:**
  - Implementation is easy
  - Building and implementing model is easy
  - Handles continuous as well as categorial data
  - Handles noise

- **Disadvantages:**
  - Minor change in data can lead to different decision trees and hence different results
  - Ineffective for small datasets

4 Data Collection

We have seven different datasets with approximately 30 records in each having multiple features. These data consist of few matches between year 1995 to 2010. All the datasets are in .xls format. First we need to convert them all to .csv and then able to load them properly.

The data [3] available is for a single team (India) against seven opponent teams. Based on these datasets,we perform the data analysis with multiple views or dimensions. This will lead us to predict the winner or the probability of the team winning the next world cup or possibly the next match. These statistics are based on the team and individual player performance in the team. So this can help us mine the labeled data for our predictions. Furthermore, we added a weather attribute to our dataset for proper prediction.

5 Prediction and classification

5.1 India vs Pak (Pakistan)

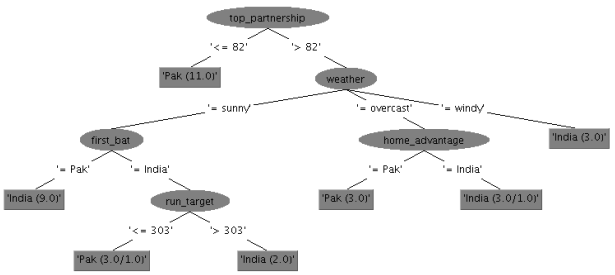

Figure 2: A decision tree using J48

- **Prediction**

  If a top partnership for India is less than or equal to 82 than Pakistan may win. Otherwise, depending on weather, if Indian team score more than 303 runs, then only India can win.

- **Results**

  - Accuracy: 94.1176 %

  - ICC world cup 2011; a second semi final match between Ind-Pak resulted in India's win. The highest partnership was 68, so very close to 82 but our prediction is not valid for this case. But if we see other half and depending on weather conditions and home advantage India should win. So half of our prediction is valid. Overall, we say average prediction but fails in the case of target run by 33

5.2 India vs Aus (Australia)

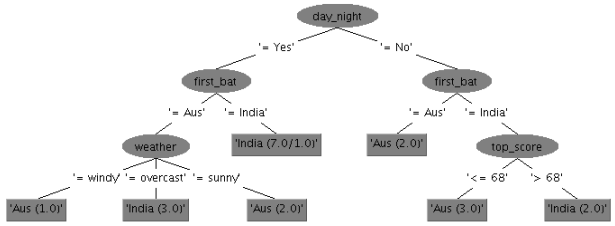

Figure 3: A decision tree using J48

- **Prediction**

  If match is Day/Night and First batting is of India then India will win otherwise it will check the weather and if it's overcast then India will win.

- **Results**

  - Accuracy: 95 %

  ICC world cup, 2011, second quarter final match between Ind-Aus resulted in India's win. It was a day night match and weather was almost overcast. So, our prediction is good.

5.3 India vs Bangladesh

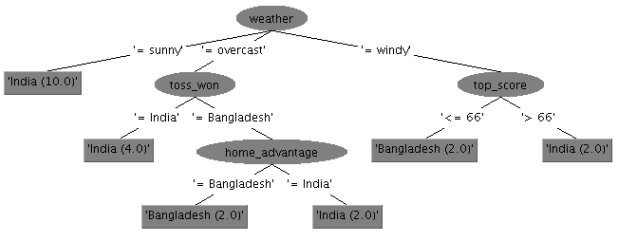

Figure 4: A decision tree using J48

- **Prediction**

  Depending on weather, if toss is won by India then India will win, even any player from Indian team score more than 66 runs chances are high for the win. Particularly India looks a strong side against Bangladesh.

- **Results**

  - Accuracy: 100 %

  - ICC world cup 2011; a match between Ind-Ban resulted in India's win. Here, the toss is won by Bangladesh, but they chose to field first. Consequently, two players from Indian team, Virat and Yuvraj scored a century. So according to our prediction, India wont the match. This prediction is most accurate one.

5.4 India vs Eng (England)

- **Prediction**

  If a match is Day/Night, then win chance are depends on home advantage. In most cases India wins. If not a Day/Night match, and Top Scorer from England team scores greater than 71 then India will lose.

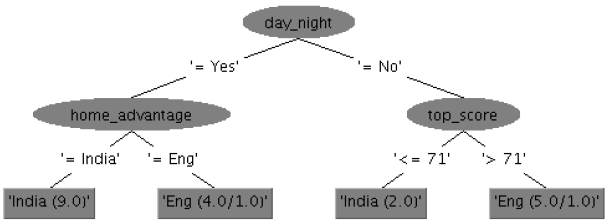

Figure 5: A decision tree using J48

- **Results**

  - Accuracy: 90 %

  - ICC world cup 2011; a match between Ind-Eng resulted in surprised tie. Here, the match was day-night and home advantage is for India, but match was tied. In such cases, our model may not get accurate result. Bur very rare chances. This kind of data we have not included in our dataset. So for this, we can say our predication is good but will not give accurate result each time.

5.5 India vs SA (South Africa)

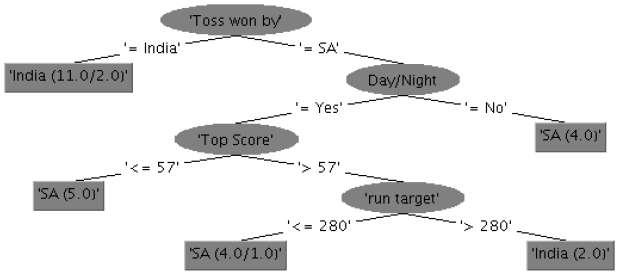

Figure 6: A decision tree using J48

- **Prediction**

  If a toss is won by India, then most of the time winning chances for India are high. Otherwise, it will check for match is Day/Night, if 'No, South Africa will win or else it will check if top individual score must be greater than 57 and target score would be higher than 280 then only India will win.

- **Results**

  - Accuracy: 88.4615 %

  - ICC world cup 2011; a group match between Ind-SA resulted in South Africa win. Even though the toss won by India and run-target was greater than 280, it results in exactly opposite of our prediction. So, we can say this prediction was not so good. But we can't judge this prediction only based on a single match.

5.6 India vs WI (West Indies)

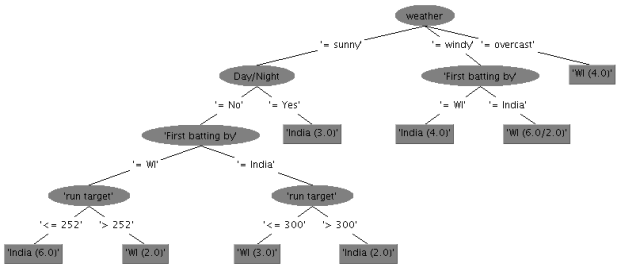

Figure 7: A decision tree using J48

- **Prediction**

  Depending on weather conditions If match is Day/Night, India will win and if India bat first and scores more than 300 runs then chances of a win for India are high while West Indies bat first and scores more than 252 runs then West Indies will win.

- **Results**

  - Accuracy: 63.3333 %

  - ICC world cup 2011; a group match between Ind-WI resulted in India's win. This was a Day/Night match and weather was sunny. so as per our prediction India won the match.

5.7 India vs SL (Sri Lanka)

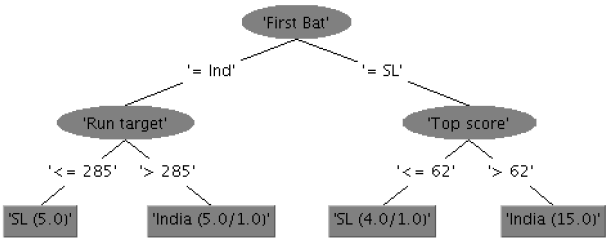

Figure 8: A decision tree using J48

- **Prediction**

  If India will bat first and scores more than 285 runs then chances of a win for India are high while Sri Lanka bat first and top scorer from Indian team scores more than 62 runs while chasing then India will win.

- **Results**

  - Accuracy: 82.7586 %

  - ICC world cup 2011; a final match between Ind-SL resulted in India's win. In this match, Sri Lanka batted first and while chasing Gambhir scored 97 runs which are more than 62 runs, so we can say our prediction is correct and this is how India won the final match of world cup 2011.

6 Application Implementation

We have developed a UI primarily to test our model. We can easily give the inputs and predict which team will win the one day match based on the given attributes. Earlier we developed C4.5 algorithm in Java and able to classify the instances. As an extension of it we built the decision tree model using Weka for each of the above prediction. Using that model we are predicting from java code which team will win the match and displaying it to UI with decision tree. As part of this project, we made a user interface for only one prediction India vs Pakistan as the match between these two team is most popular in the world of cricket.

Languages Used: Advanced Java, Spring MVC framework 3, JSP, CSS, Java Script

When, we run the application, home page will be available as below:

Figure 9: A user interface home page

In our application, we have used Java Script to apply all the field validation like only number is accepted for particular attribute, only few values are accepted. This can be notified to user via Java Script pop up when he click on the predict button as shown below.

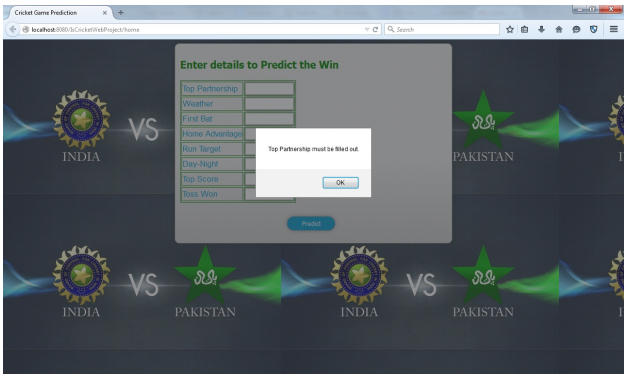

Figure 10: A user interface home page

In the final phase of UI, we are showing the prediction which team will win and the decision tree for the same. This whole prediction happens at Java back-end. In the back-end it will scan all the input values and classify using our saved Weka model and give the predicted win result for the given test instance. The final web page is show below.

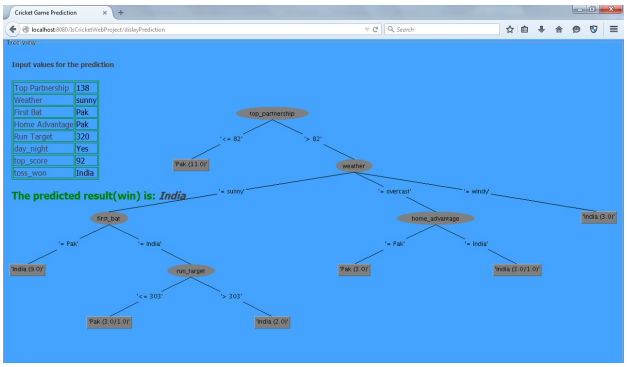

Figure 11: A user interface home page

7 Challenges Faced

- To find the correct dataset.

- We have to use Weka due to need of .model file for testing our data from UI. So, we had to rerun all the steps in Weka again.

- The attribute "weather" was included later, which was a difficult one to get for a particular city and time and day.

8 Lessons Learned

- First step would be to decide correct dataset.
- We have to handle 'NA' values which might strongly affect the classification.
- Sometimes, even though the higher accuracy, match result may be negative same as the case for India-SA match prediction in our case.

9 Future Work

- This application can be further extended for all seven opponents. Taking the dataset for all teams and finding details between all the tournaments, we can predict the result for any team against anyone.
- Currently we are testing our prediction only for a match between India-Pak. We can enhance our application to test all of the predictions made using decision trees in Weka.
- This model can be updated for other sport matches and leagues.

10 Conclusion

Cricket prediction in almost all the cases is good. Our predictions, based on the data from 1995 to 2010, were true for almost all the matches except the one against South Africa in the ICC Cricket World Cup 2011. Including weather, was a good decision. It displays more accurate results and is the important factor in all decision trees as cricket is an outdoor game. For the future, we can enhance our application for all the teams and predict any tournaments. Even we can advanced this application for any sport.

References

[1] Ananda Bandulasiri, "Predicting the Winner in One Day International Cricket," *Journal of Mathematical Sciences & Mathematics Education*, Vol. 3, No. 1., 2008.

[2] Michael Bailey & Stephen R. Clarke, *Predicting the Match Outcome in One Day International Cricket Matches, while the Game is in Progress,* The 8th Australasian Conference on Mathematics and Computers in Sport, 2006.

[3] *Predictive analytics in cricket, Simafore LLC.* 2014
`http://www.simafore.com/predictive-analytics-cricket-statistics-data`

[4] J. R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, 1993.

[5] J. R. Quinlan. "Induction of Decision Trees," *Machine Learning,* 1(1):81106, 1986.