Predicting the outcome of ODI Cricket matches

Samiran Roy 153050091
Sushant Shambharkar 153050081
Ganesh Bhambarkar 153050072
Surendra Singh Rao 153050069

February 2016

Abstract

Cricket is a very popular game not only in India but also in the world. Every team has one ultimate goal that is Victory in the match. ODI cricket match results depend upon various factors, related to scoring as well as physical strength of the two teams. The complex rules surrounding the game, along with the numerous parameters affecting the game, including but not limited to cricketing skills and performances, match venues, toss, weather conditions present significant challenges for accurate prediction. Our task is to investigate to what degree we can predict the outcome of cricket matches, specifically ODI Matches. This problem has been well investigated for games like basketball and soccer, but yet to be researched for cricket. To solve this problem we have used SVC, a Machine Learning Technique which is used for classification problems. This prediction technique can be used for any ODI match between any top teams which play cricket. Our results show that more training datasets and number of affecting attributes give better accurate results.

1 Description

Our task is to investigate to what degree we can predict the outcome of cricket matches, specifically ODI Matches. Given the popularity of this multi-million dollar industry, there is a strong incentive for match fixing and underground betting. The complex rules surrounding the game, along with the numerous parameters affecting the game, including but not limited to cricketing skills and performances, match venues, toss, weather conditions present significant challenges for accurate prediction. This problem has been well investigated for games like basketball and soccer, but yet to be researched for cricket.

2 Dataset

The main challenge which took up most of our time, was integrating data from 3 sources. The sources are described below:

2.1 Match data

We will be using the dataset from `cricsheet.org`

The data-set provides ball by ball data for matches. The results of the games can be a *win*, *tie* or *no result*. Each YAML file contains data about one game. In case of *no result*, the reason is not provided. Although the *city* where the games took place is present in most files, about 10% of files do not specify the city. We'll need to manually fill the city by looking at the venue in that case. Some matches span for 2 days. We'll consider only the starting date as our feature for prediction. The data of total 1,164 ODI matches are present.

2.2 Weather data

`https://www.wunderground.com/history/index.html` provides daily weather data. We can query the data for any city on a particular day using a simple URL change. Then we can parse the information from the website. This allows us to automate the gathering of the data per city, per day and put it as features in our data.

For example, this link gives the weather data for Mumbai on 21 March 2006: `https://www.wunderground.com/history/airport/VABB/2006/3/21/DailyHistory.html?req_city=Mumbai&req_state=&req_statename=India&reqdb.zip=00000&reqdb.magic=1&reqdb.wmo=43003`

The website provides following information which is relevant to our project:

- Humidity
- Wind Speed
- Temperature
- Dew Point

2.3 Handcrafted features

We handcrafted three features: Win/Loss Ratio, Home field advantage, Venue Win Rate by manually copying from *statsguru* . These features are described in the methodology section.

Methodology

A significant number of important features are manually copied from the website, using data from *statsguru*, therefore we had to reduce the size of the dataset. We removed all matches not played by India. We were motivated by the following reasons:

- Making it feasible to handcraft features

- India has played several matches against all the teams in this world. This makes for efficient samples for training.
- A lot of games are played in Indian pitches, providing us accurate pitch data

We only focused on Autralia, New Zealand, Pakistan, Sri Lanka, West Indies, England, Bangladesh, South Africa as opponents, where matches are "interesting" and hard to predict. Matches like India vs Ireland or India vs Afghanistan are easily predictable, removing these samples lowers the accuracy of our classification.

We take match data from 2005-2016. It is better to recent results of cricket while modelling the game, since some teams change very rapidly, and past data serves only to increase noise in the model.

We also remove all matches ending in a draw or no result for obvious reasons.

If we want to do better than a coin toss, we first have to take the winning rate against a team as a feature. Apart from that, match specific features are used like the condition of the pitch, the players playing the game, the venue, the toss, the strengths of batting and bowling. We use the following features in our dataset.

From *cricsheet*

- Venue [Categorically encoded]
- The team playing against india [Categorically encoded]
- FirstToBat : 0, if India bats first
- All the 22 players playing the game [Categorically encoded]
- The winning team (Which we have to predict)

From *Wunderground*, the pitch conditions of the area

- Humidity
- Wind Speed
- Temperature
- Dew Point

Hand-crafted features from *statsguru*

- Win/Loss Ratio: Taken from past history of India against every team - given India is batting/bowling

- Home field advantage
- Venue Win Rate

Since a lot of the features are categorically encoded, we end up with a sparse matrix of close to 1000 features.

Results

We use Decision Trees, Support Vector Classification, Logistic Regression, and Naive Bayes of scikit-learn library in python. We tune the hyperparameters for each using grid search, and report results for 3 fold cross validation on a combination of the above features:

Table 1: Results obtained using 3-fold cross validation

Method	Accuracy
Naive Bayes	0.725
Logistic Regression	0.693
Decision Tree	0.734
SVC	0.75

RMSE(as used in [1]) was used by us for comparison. The best accuracy was given by Support Vector Regression, using an rbf kernel.

We beat the state of the art result of RMSE: 0.593[1] by training on merely 120 samples, as opposed to all the odis since the 1970's[1], using the above described features.

Future Work

There is no dearth of Data in cricket, but it is distributed and hard to access. The most important thing that needs to be done is to create an api, that interfaces *cricsheet*, *statsguru* and *cricmetric*, and can query the database. Then we use extra features, like batting/bowling averages of every player against teams, modelling player weaknesses, Day/Night match, captainship. And use more samples for training. We also have not exploited the confidence bounds of data. We also have not preprocessed the data thoroughly according to each classifier.

Conclusion

This project makes several contributions. We first parse YAML data from cricinfo.org ans extract the basic features given in 1

We improve over the state of the art scores in cricket classification, mentioned in [1] using more features and minimal training data. Weo removed the

"easy" samples from our data and only trained for 120 samples. We show that cricket matches can be predicted with high accuracy, using carefully constructed features. The problem lies in getting training data for cricket. There exist several websites where we can look for features manyally, but no unified approach to query them.

Literature Survey

[1] Amal Kaluarachchi and Aparna S Varde. Cricai: A classification based tool to predict the outcome in odi cricket. In *Information and Automation for Sustainability (ICIAFs), 2010 5th International Conference on*, pages 250–255. IEEE, 2010.