CS 578 Project: Effect of Team Selection on Performance

Parameswaran Desigavinayagam Pradeep Kumar Srinivasan

Problem: Given the composition of two cricket teams along with past statistics of the teams and players, how well can we predict their match performance?

Dataset: The dataset consists of YAML files with information about the T20 cricket matches (from https://cricsheet.org/). Each file has ball-by-ball data (batsman, bowler, non striker, runs scored etc) for matches played within the Indian Premier League.

Experimental Setup:

- We will extract individual statistics like average, strike rate, team records, etc. for each player on either side and use them as input features, along with other team- and match-related features.
- We wish to predict the match outcome - win or loss.
- We will use feature selection to narrow down the features. (For example, we expect that the batting averages of the bowlers and the bowling averages of the batsmen will be of little importance.)
- We can compare different feature selection algorithms - greedy selection, forward fitting, and L1 norm regression.
- We will use slack variables in order to possibly lessen the effect of outliers like rare extraordinary performance by players which may overshadow the effect of other variables.
- We will compare classifiers like linear SVM, kernel SVM, logistic regression and ensembles.
- To set the hyper-parameters for our model, we will use k-fold cross-validation. And to test our accuracy, we will use precision and recall, and plot them against number of samples and also regularization parameter.
- We will run it on different subsets of the dataset, one season at a time, or a certain percentage of matches from all seasons, or finals vs league matches.
- We will test the validity of our model with data from other T20 leagues whose data is also available.
- Our project will be in Matlab and Python.

Progress

We have implemented a complete workflow (with some part or algorithm in each step) consisting of

- Aggregation (player statistics - strike rate, batting average, bowling economy of each player)
- Feature selection (Greedy Subset selection)
- Classifier (Primal and Dual SVM)
- Hyper parameter optimization with K Fold cross validation
- Testing on the remaining unseen data

Instructions to run

- **Requirements:** python-numpy, python-scipy, python-yaml
- **Feature aggregation (Python)**
  - python3 features/extract_features.py
  - Or, set the path for python3 and run ./run-classifier.sh
- **Classification (Matlab)**
  - cd into matlab folder
  - Run start(<percent training>, <number of folds>, <feature subset size>)
  - start(0.70, 5, 5)

Approach

- Since, strike rate and batting average of players are the most commonly used statistics in cricket, we ran our basic version using only these features.
- Our dataset has YAML files for all matches in the 10 seasons of the Indian Premier League. We parsed the files and aggregated the individual statistics for players in each match based on all the matches played up to the current match.
- For now, we train and test on matches from one season (with all past seasons providing player statistics).
- The current features are
  X: [number_of_matches, 22 * k] y: a vector of size number_of_matches
  (k is the number of statistics for each player)
- We split the data into (Xtraincv & ytraincv) and (Xtest & ytest)
  (where number of rows in traincv = training_percent * number_of_matches)
- We run Greedy Subset selection to get a subset of features.
- We now train and test two classifiers: linear primal SVM and dual SVM.
- For Dual SVM
  - Perform Hyperparameter estimation during cross-validation to get the best value for C & Gamma (Kernel parameter)
  - Try pairs of values of C & Gamma by iterating over [0.001, 0.01, 1, 100, 1000] for both values and find best values which maximizes the accuracy on running k-fold CV with the parameters.
  - Finally test our model on Xtest, ytest using the optimum parameters found above.

Experimental Results:

Using the strike rate of the 22 players (11 x 2 teams) as features, we got the following accuracy:

1) Running it on per season data

S.No	Season	Dual SVM Accuracy
1	2	0.3889
2	3	0.3333
3	4	0.4545
4	5	0.5217
5	6	0.5652
6	7	0.4444
7	8	0.6111
8	9	0.3889
9	10	0.5556

2) with different percent of data of whole input for training (All seasons) (with just strike rate of players)

S.No	Percent of data trained on	Dual SVM Accuracy (Strike rate of 22 players)
1	95%	0.4483
2	90%	0.4483
3	85%	0.4368
4	70%	0.4368
5	75%	0.3534

3) Accuracy with different subset of features

S. No	Percent of data trained on	Size of Feature Subset	Dual SVM Accuracy with different features		
			(Strike rate of 22 players)	(batting average of 22 players)	(bowling economy of 22 players)
1	70%	5	0.4335	0.4335	0.4335
2	70%	10	0.4335	0.4335	0.4335
3	70%	15	0.4335	0.4335	0.4335
4	70%	22	0.4220	0.4335	0.4335

4) with different feature subset size (additional features)
Features: Strike rate and average of 22 players (of two teams)
1-11 strike rate of first team's players
12-22 batting average of first team's players
23-33 strike rate of second team's players
34-44 batting average of second team's players

S.No	Percentage of data trained on	Size of Feature subset	Dual SVM Accuracy
1	70%	5	0.4335
1	70%	11	0.4335
2	70%	22	0.4335
3	70%	33	0.4277
4	70%	44	0.4220

7) However, with Primal SVM, we got the error "The problem is infeasible." We hope to investigate that problem soon.

Remaining Work

As mentioned in the project proposal, we will implement the remaining classifiers (like ensemble classifiers) and compare them on precision and recall, and generate other plots. We will also compare other feature selection algorithms.

The current set of features give us no better results than random chance. So, we plan to test with the other features we have extracted from the data, such as bowler statistics, team win

rate, team net run rate, etc. Also, we need to investigate why the accuracy is the same even for different features. We will manually test our model on some inputs to ensure the above and interpret the results to see if there are any issues with the features we added. Another issue is that of player order in the training set. Right now, we have players in alphabetical order. It would make more sense to arrange them in the playing order. We also plan to experiment with derived features such as average + strike rate that are commonly used in cricket.

Also, feature selection does not give us any improvement in performance with the current datasets, because we have only tested one type of feature at a time. So we plan to combine different features like strike rates, averages, etc. in the same training set, and also train and test across seasons so that we have more data points.