

Prediction of winner in NBA basketball games

Zejian Zhan

Outline

- Motivation
- Data extraction
- Model and Data analysis
- Feature analysis
- Algorithms and results
- Conclusion

Motivation

- Predicting the winner in a game based on the previous performance is very valuable to business managers, coaches, players, fans, gamblers, and statisticians alike.
- Try to find the better algorithm for this prediction with box scores

Data Extraction

Tue, Oct 28, 2014	10:30 pm	Box Score	Houston Rockets	108	Los Angeles Lakers	90
Tue, Oct 28, 2014	8:00 pm	Box Score	Orlando Magic	84	New Orleans Pelicans	101
Tue, Oct 28, 2014	8:00 pm	Box Score	Dallas Mavericks	100	San Antonio Spurs	101
Wed, Oct 29, 2014	7:30 pm	Box Score	Brooklyn Nets	105	Boston Celtics	121
Wed, Oct 29, 2014	7:00 pm	Box Score	Milwaukee Bucks	106	Charlotte Hornets	108 OT
Wed, Oct 29, 2014	9:00 pm	Box Score	Detroit Pistons	79	Denver Nuggets	89
Wed, Oct 29, 2014	7:00 pm	Box Score	Philadelphia 76ers	91	Indiana Pacers	103
Wed, Oct 29, 2014	8:00 pm	Box Score	Minnesota Timberwolves	101	Memphis Grizzlies	105
Wed, Oct 29, 2014	7:30 pm	Box Score	Washington Wizards	95	Miami Heat	107
Wed, Oct 29, 2014	8:00 pm	Box Score	Chicago Bulls	104	New York Knicks	80
Wed, Oct 29, 2014	10:00 pm	Box Score	Los Angeles Lakers	99	Phoenix Suns	119
Wed, Oct 29, 2014	10:30 pm	Box Score	Oklahoma City Thunder	89	Portland Trail Blazers	106

8	105	16	94
20	105	18	86
4	88	21	87
2	104	11	106
12	89	25	97
9	95	3	96
14	104	5	96
17	115	16	93
4	91	18	99
6	79	7	91
25	104	23	100

Copy & paste to Excel and write a macro to convert string of team names to integer value ranging from 1 to 30

Naïve way for prediction

- We can compute the win percentage of both teams, and choose the one whose percentage is higher
- The result is like:

season	accuracy
2011-2012	0.6363636363636364
2012-2013	0.6403580146460537
2013-2014	0.6455284552845528
2014-2015	0.647479674796748
2015-2016	0.6304559270516718

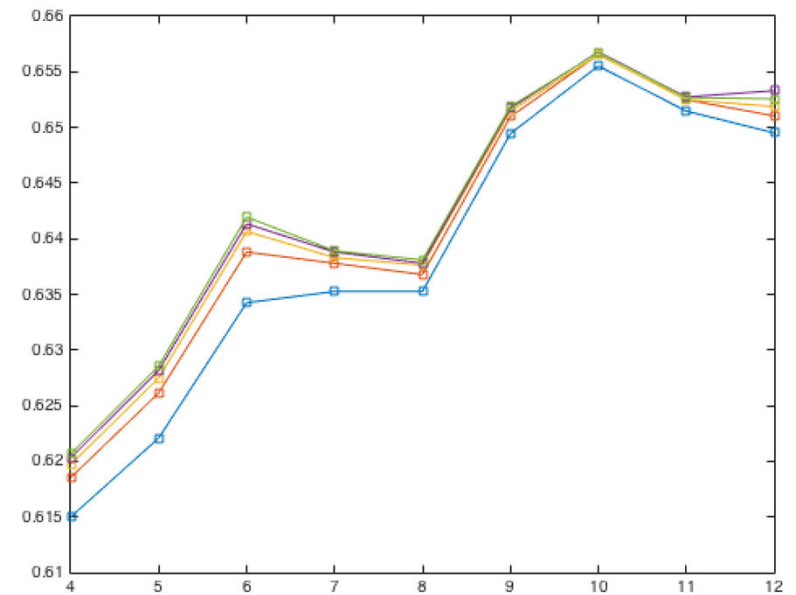
Average should be around 64%.

Features

- Win percentage should be taken into consideration
- Average point differential is a good way to show the strength of a team
- We should also look at the recent performance of teams:
 - win percentage in recent N games
 - average point differential in recent N games
- Win percentage as a visiting team and as a home team

Features analysis

	A	B	C	D	E
2011-2012	0.6303	0.63141	0.64715	0.65854	0.6464
2012-2013	0.64646	0.64524	0.64146	0.66016	0.64944
2013-2014	0.63535	0.65094	0.64065	0.64065	0.65552
2014-2015	0.63838	0.64687	0.63984	0.64878	0.63121
2015-2016	0.63636	0.62897	0.62602	0.63415	0.64843
AVG	0.63737	0.64068	0.63902	0.64846	0.6462

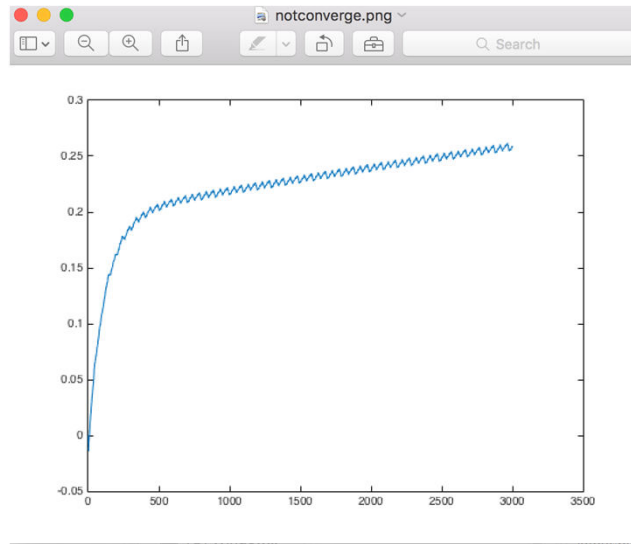


SVM

- Using existing function call in matlab with linear kernel

2011-2012	0.6846
2012-2013	0.6724
2013-2014	0.6813
2013-2014	0.6556
2014-2015	0.6713
Avg	0.67304

Linear regression



Not converged
after huge iteration

PCA

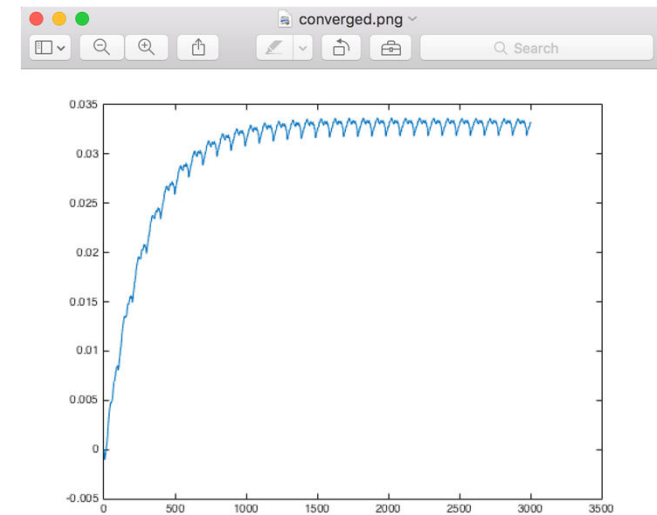
- Reduce the dimension and boost the convergence

latent =

190.0989
181.5020
87.0034
66.8518
48.5988
35.6655
4.5354
4.3422
0.0923
0.0412
0.0308
0.0126
0.0093
0.0067
0.0060

ans =

0.3072
0.6005
0.7411
0.8492
0.9277
0.9853
0.9927
0.9997
0.9998
0.9999
0.9999
1.0000
1.0000
1.0000
1.0000



Result of linear regression

2011-2012	0.6789
2012-2013	0.6942
2013-2014	0.6491
2014-2015	0.6718
2015-2016	0.6522
Avg	0.66924

Multi-layer Perceptron

- α takes 0.1 0.3 0.6 0.9 and momentum takes 0.25 0.50 0.75

rate	0.25	0.5	0.75
0.1	66.9084316	66.81777	67.18042
0.3	66.9386522	67.02931	67.48262
0.6	66.7573285	67.57328	67.69417
0.9	65.0045331	67.4524	64.49078

One layer with four neurons

rate	0.25	0.5	0.75
0.1	66.9386522	66.72711	66.69689
0.3	66.7875491	67.42218	66.72711
0.6	67.0897552	67.51284	67.4524
0.9	67.7848293	67.51284	65.27652

One layer with eight neurons

rate	0.25	0.5	0.75
0.1	66.9386522	66.90843	66.75733
0.3	67.0595346	66.84799	67.36174
0.6	67.3919613	67.4524	67.33152
0.9	67.3617407	67.21064	63.58416

One layer with twelve neurons

Too few hidden units will generally leave high training and generalization errors due to under-fitting. Too many hidden units will result in low training errors, but will make the training unnecessarily slow, and will often result in poor generalization

‘Optimal’ pick

Two hidden layers with eight and twelve neurons for each layer respectively

season	accuracy
2011-2012	0.6917
2012-2013	0.6872
2013-2014	0.6915
2014-2015	0.6899
2015-2016	0.6877
Avg	0.6896

Conclusion and future works

- All algorithms got greater than 64% accuracy
- The Multilayer Perceptron with two hidden layers got the best accuracy: 69%
- Need to import teams' detailed statistics by using SQL to store all players' statistics and executing queries for obtaining better and detailed features to measure teams' performance