Predicting Kobe's Performance

Stat 154 Modern Statistical Prediction and Machine Learning

Cheng Dong, Min Jo, Frank Zhang

Ridge, Lasso, LAR, and Forward S

INTRODUCTION

Basketball was invented in 1891 by Dr. James Naismith. At first, the game was very simple. The baskets were made from peach baskets, and the ball in use was actually a soccer ball. Over 100 years later, the game grew to become much more complex with innovations such as the three-point line, a replay system, and of course, comprehensive statistics.

Kobe Bryant is currently one of the top scorer's in the NBA, and one of the greatest of all time. He is the youngest player to score 31,000 points, is currently 4th in all-time scoring, and has won two scoring titles. What defines Kobe Bryant is his undeniable ability to put the ball in the hoop.

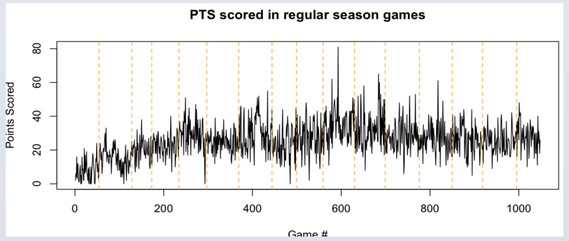
Therefore, one question we want to investigate is how Kobe's prior performances affect his next coring performance. In order to answer that question, we will predict Kobe Bryant's scoring performance for the next game he plays by using machine learning techniques.

EXPLORATORY

We first found the correlation between predictors and points. We found that PTS is correlated with parameters such as game starter, minutes played, field goals, field goals attempted, average points from past 5 games, and GmSc (a measure of Kobe's productivity during a game).

Game Starter	Minutes Played	Field Goals	Field Goal Attempts	Average Points	Game Score
0.462004	0.485752	0.491646	0.519255	0.509064	0.456355

Since we had many predictors, we wanted to check the correlation values between our predictors. We found **86** pairs of variables with correlation higher than 0.5 and **37** pairs of variables with correlation higher than 0.7. This large number does not even include collinearity between linear combinations of several variables with another variable. Thus, we decided to explore many models that performed variable elimination and dealt with possible collinearity.



The above plot shows the distribution of the number of points Kobe scored over time. Larger game numbers indicate later games.

DATA

We gathered seasonal data from the basketball-reference website. We built table for Kobe's personal performance data from years 1996 to 2012. Since we wanted to predict Kobe's performance for his next game, the opponent he is playing with is an important determining factor for his performance. Thus, we built tables of offensive and defensive data for Kobe's opponent team from years 1996 to 2012. We then merged Kobe's personal data with his opponent data by combining on the opponent team's name.

Since the parameters describing each game is unknown prior to the game, we used a 5 game grouping scheme to create predictor variables. We took the mean of the statistics from the previous five games of the game we want to predict and used those as predictor variables. At this point, we have 1064 games and 57 variables.

We had a few problems within our data that made certain entries unusable for model building. We had to use regex to convert some of the parameters into workable formats. These parameters included age, time, home or away games, win/loss, etc.

Some other problems we experienced included dealing with NAs and Infs in the Kobe's personal data. These resulted in percentage predictors. For instance, field goal percentage is equal to field goal successes over field goal attempts and will yield in NA if field goal attempts is zero. Since the percentage predictors were already represented by the variables that yielded in the percentage, we decided to remove these columns due to redundancy and obvious collinearity. Our final data set consisted of 1048 games and 51 variables.

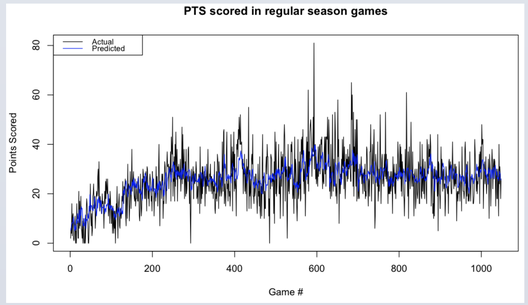
METHODS

Cross validation helped us evaluate the different models we used. Since we have a set amount of data, it was very useful for us to be able to use different training sets to see how well our model really performed.

OLS & GLS

Method	CV Error	Training	Testing
OLS (Forward Selection)	6.628941	6.943838	6.74374
OLS (Backward Selection)	6.490223	6.970129	6.6932
GLS (Forward Selection)	6.950420	6.797303	6.805020

In order to predict Kobe's points in the next game, we utilized forward/ backward stepwise method to reduce dimensions of the data by selecting predictor variables. Using the OLS method with these variables, we actually got the lowest error of the all methods which is 6.49. The error is the mean of absolute values of the differences. For GLS, the error was 6.95. It should be noted that we used variable selection techniques to prevent overfitting.



The above plot shows the distribution of the number of points Kobe scored over time. Larger game numbers indicate later games.

PCR

K	CV Error	Error (standardized)	Training	Training (stand)	Testing	Testing (stand)
1	6.5736	7.180537		7.16173		7.0078
5	6.5636	7.177982		7.14285		6.99745
10	6.4981	7.126609		7.06924		6.96534
40	6.4023	7.128349		6.89359		6.90954

The objective of our principal components analysis was to find a linear transformation of a set of our 50 predictor variables into a new set denoted by P, where the new set has certain desirable properties.

1) The elements of P are uncorrelated with each other in the sample

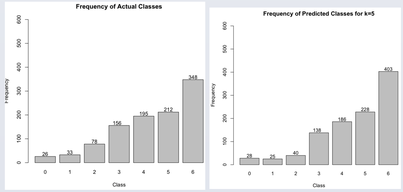
2) Each element of P accounts for as much of the combined variance of the x's as possible, consistent with being orthogonal to the preceding p's.

As a result, we noticed that, as n increases, the magnitude of differences (errors) decreases.

KNN

```
k = 1 error rate 0.7509542
diff in pred classs -1 4 3 -9 6 22 -25
k = 5 error rate 0.7251908
diff in pred classs 3 -12 -35 -13 -20 14 63
k = 7 error rate 0.7041985
diff in pred classs 2 -19 -48 -12 -14 -9 100
k = 10 error rate 0.6965649
diff in pred classs 3 -21 -54 -34 -54 6 154
k = 15 error rate 0.6832061
diff in pred classs 2 -24 -43 -57 -50 -19 191
```

Points	Class
0-4	0
5-9	1
10-14	2
15-19	3
20-25	4
25-30	5
30+	6



With KNN, we tried several K's, and found that as *k* increased, the overall error rate went down. However, after further investigation, the reason why the error rate went down was because the algorithm predicted more and more 6's. The number of 6's correctly predicted went up at a faster rate than the rate for the number of 1-5's correctly predicted went down. Therefore, the error rate decreased even though the predicted values shifted towards more 6's.

We decided that the best *k* for this dataset is actually 5 because while the error went down as *k* increased, it was for the wrong reason. The algorithm almost seemingly blindly predicted more and more 6's.

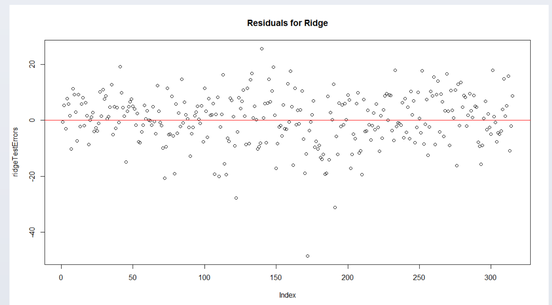
Benchmarks

One benchmark that we used was simply looking at the previous season's scoring average as our prediction for the next game. Therefore, we could not predict any game for the first season and our predictions are the same for all games in a season. The average absolute value of the residual was 8.485.

Another benchmark that we used was using his running career average as the prediction. This means that our prediction was very volatile early in his career and becomes stable towards the end of his career. The average absolute value of the residual was 8.191 which meant that it id better than the other benchmark, but still worse than our machine learning techniques.

Method	CV Error	CV Error (standardized)
Lasso	6.570095	7.280089
Ridge	6.178731	7.074153
Forward Stagewise	7.701985	NA
LAR (Least Angle Regression)	6.725991	7.326344

Method	Training	Training (stand)	Testing	Testing (stand)
Lasso	6.72508	6.734532	7.515963	7.177202
Ridge	6.785631	7.060684	6.819793	6.873079
Forward Stagewise	7.701985	6.785631	7.060684	6.873079
LAR (Least Angle Regression)	6.716766	6.732922	7.16425	6.968429



While ridge regression is used to deal with high dimensional data, this method smoothes the coefficients of the model out instead of setting them to zero and completely eliminating its effects. The pros of this model is that each parameter will always have some, no matter how small, effect on the response variable. This pro is also a con since this makes it impossible to ever completely rid of collinearity. Overall, ridge regression performed the best among the LARs-like models. The residuals are randomly distributed about zero.

Conclusion

In order to predict Kobe's scoring performance for the next game, we tried multiple methods. It is difficult to distinctively point out the best method because most methods yielded similar results. However, after completing this project, there are several improvements that could have affected our results.

-More predictor variables (Kobe's one-on-one defender, his past performances specifically against that team. Etc.…)

-Apply our techniques to more players in order to better understand the accuracy of our methods

-

However, based on our predictions, we can see that most algorithms predict that the average points Kobe will score is around 26 points, which is consistent with his career average. The average residual is around 6.8 which is not terrible and considerably better than any benchmarks.

Acknowledgements

Johann Gagnon-Bartsch, Dat Duong, Kobe "Black Mamba" Bryant