Kobe Bryant Shot Selection

Taufer
PSTAT 231, Spring 2016
Colin Menz, Shaoyi Zhang, Emily Zheng

Question

In recent years, the United States has seen a boom in the sports betting market. Thanks to the popularity of fantasy sports, Americans find themselves engrossed in the world of sports. Competitors in traditional fantasy leagues follow their players for the whole season. Fans of the season-long format have welcomed the newer daily fantasy format. With daily fantasy, a competitor selects a team and they win if the team outperforms most of the others. Competitors have the option to swap underperforming players the following week with no consequences.

Teams are scored on their real-life performance. A team's players earn points based on their actual game statistics. Exceptional performance in particular statistics can be worth more. Competitors focus on players which benefit this goal, thus the prediction of future actions of an athlete is of interest in fantasy sports.

Specifically we want to predict whether or not Kobe Bryant successfully makes his shot. For his entire 20-year NBA career Kobe Bryant played on the Los Angeles Lakers. From 1996 to 2016 Bryant attempted over 30,000 field goals. We are investigating which variables are most relevant to making two- and three-pointers, then building a model to predict the success or failure for a test data set of 5,000 records based on these findings. We will seek trends in the data and discuss whether or not our findings coincide with conventional wisdom.

Data

For this analysis, the original dataset has every one of the 30,697 shots attempted in Kobe Bryant's NBA career - it was found on Kaggle. It contains the locations and circumstances of each goal attempted, as well as whether or not the shot was successful ('Shot_made_flag'). In 5,000 random records the 'Shot_made_flag' is removed so we are unable to use these records if we would like to self-evaluate our model without external information from Kaggle. Because these 5,000 records were randomly selected they are well spread out, thus when they are removed there are no large holes of missing time in our data. Our updated data set now consists of 25,697 records which we further split into a training set and a testing set. Each record contains information for 25 variables, ranging from type of shot to time left in the period.

The original data set has 15 factor variables and 10 numeric variables. This includes information such as location, circumstances, actions, and timing. We will add or remove variables to the original data set because the existing variables might be highly correlated or the analysis might benefit from using a dummy variable.

Factors variables 'Team_id' and 'Team_name' were immediately deemed unuseful - Bryant has only played on the Los Angeles Lakers. To isolate information of whether or not the game occurred at home or away, we used information from the variable 'Matchup' to create the indicator variable 'Home', which is 1 when the game is at home and 0 otherwise. We did not find numeric variables 'Lat' and 'Lon', which represent the player's position on the globe, to contribute significantly to predictions because the key information is whether or not this attempt occurred during a home game. 'Minutes_remaining' and 'Seconds_remaining' were joined into the new numeric variable 'Seconds' which is the number of total seconds remaining in the current period at the time of the shot attempt.

The factor variable 'Action_type' has over fifty possible categories, which can lead to complications down the road. To decrease the number of categories we combined this variable with the similar variable 'Combined_shot_type' into the factor variable 'Type'. 'Combined_shot_type' is a less specific description of the type of shot made, therefore it has fewer, broader categories than 'Action_type'. We replaced categories of 'Action_type' with less than 50 observations with its corresponding 'Combined_shot_type' classification.

Attempted Approaches

There are many different approaches that we considered; however, it became clear that many of them would not be suitable for our analysis. We first thought of principal component analysis as a way to choose variables because it is able to produce a low-dimensional representation of the dataset. But our data contains both continuous and factor variables, which is not suited for principal component analysis as many of our categorical variables contained vital data. In addition, PCA attempts to find a representation of the observations that explains the variance, whereas we are concerned with finding a probability.

The next approach we tried was classification using trees. First we produced a single tree in order to get a sense of which variables might be important for prediction. It quickly became clear that this would not be sufficient for analysis, as it was fairly easy to interpret but was perhaps too simple and lacked the prediction accuracy that we could achieve with bagging or random forests.

In terms of modeling our predicted variable, we also considered linear discriminant analysis, which is useful for well-separated classes. But this approach was also deemed inferior to logistic regression due to the fact that we had only two response classes. Linear discriminant analysis works well when n is small and the distribution of predictors is approximately normal, but our data had neither of those things. We had roughly 25,000 observations and it does not make sense to assume normality, so the idea of using LDA was scrapped.

It is worth noting that cluster analysis would not be particularly useful for us as our goal is to predict rather than separate. It would not make much sense to have two clusters for made and missed shots, so we decided upon logistic regression in order to obtain an easily interpretable probability.

Final Approach

Because our dataset contains both continuous and categorical variables, we use random forest to identify relevant predictors. Random forest is a very efficient statistical learning method. It works by building many decision trees on a bootstrapped training sample. While building these trees, a random sample of possible predictors are considered at every split. Using this method we can identify strong predictors in our dataset.

With random forest we generated 500 classification trees. The key result from using the random forest method is the measure of 'mean decrease Gini' for each predictor variable in this dataset. Variables with larger decreases in this measure play a greater role in partitioning data into defined classes. Using this measure we can pinpoint variables which help us identify trends in Bryant's performance. We select the independent variables 'Opponent', 'Type', 'Seconds', 'Loc_x', 'Loc_y', 'Game_date', 'Season', 'Shot_distance', and 'Period' based on the cut-off value of 450 for the measure of mean decrease in Gini. These variables will be used to build a predictive model.

The dependent variable we are predicting is 'Shot_made_flag' which is an indicator that has '0' for a missed shot and '1' for a successful shot. Because there are only two possible outcomes we will build a multiple logistic regression model that gives us an interpretable probability for a binary outcome.

The multiple logistic regression model is a nonlinear transformation of the linear regression model. It is generally used to study the effect independent variables have on the probability of obtaining a particular value of a binary dependent variable. The goal of our analysis is to specify an equation which predicts the probability of a successful shot as a function of existing circumstances.

Results

As exploratory data analysis, we investigate Bryant's field goal percentage. Bryant's career starts off with a clear upward trend. In the 2000-01 season there is a local peak, which is unmatched until his career apex of 48.8% during the 2007-08 season. Between these seasons there is a period of relatively low success rates, which is reminiscent of the start of his career. After his career peaked in 2007-08 at age 29 there is a

downward trend in performance until his retirement in 2016. Due to aging, it makes sense that Kobe's performance falls off after 2007, when he was in top physical condition.

From what we observed through the random forest method, we identified nine significant variables for building a model. We ultimately settled on the multiple logistic regression model, because our dependent variable is binary.

If each variable is zero, our model assumes the shot has a 99.7% probability of success. This scenario is impossible, because there are uncontrollable circumstances surrounding every attempt. Accounting for those, we identify situations that can significantly alter our prediction. Positive coefficients help to increase the probability that the shot was successful while negative coefficients decrease that probability.

It appears that the type of shot attempted has a significant impact on Kobe's probability of making any given shot. All types of dunks, which are field goals typically slammed into the net, have high coefficients and therefore increase his chance of making the shot. On the other hand, jump shots and fadeaways are among the most difficult shots to make in basketball and our model indicates that Kobe generally has a tough time getting them to fall.

Kobe also appears to perform poorly against certain teams in the eastern conference such as the Brooklyn Nets, but in a typical season he generally plays only two games per year against eastern conference teams which may affect our results.

Considering just the 'Season' variable, there is a noticeable upward trend in success rate as years pass. It is unclear why this occurs. Based on the observed percentage of shots made by season, as illustrated in the appendix, we expect a cyclical trend rather than a linear one. Looking further by each game, we notice a decreasing success rate as each period in the game passes. Again, it is not clear how this trend in our model reflects his observed performance.

For random forest model, we achieved test error rate 31.72%. By setting a different seed, the test error rate fluctuated around 30%-35% on randomly partitioned data sets. For logistic regression model, we achieved a test error rate of 31.65% and it will fluctuate around 30% ~33%.

Conclusion

We used logistic regression on the data set without variable selection. The test error rate was relatively high compared to our final model. This justifies our selection of predictors. Using the training data on the random forest method we achieve a low error rate of 0.000476 while we have a much higher error rate of 0.348 for the test data. This suggests overfitting of the data.

We can mitigate this issue in a number of ways. One method is to decrease variance by generating more trees in our random forest. By doing so, our expected variance will decrease as the square root of the sample size. However, we tried to increase the number of trees from 500 to 1000. The error rates are almost the same, while the program became much slower due to too many trees.

We could also reduce the number of variables in the random forest to reduce overfitting. This involves specifying the number of predictors in each tree within the random forest algorithm. Using this approach the test error rate decreased, although the training error rate increased. This suggests including fewer variables will reduce overfitting.

Despite our best efforts, there will always be random chance underlying every shot attempted in every basketball game. We found with so many factors acting at once, it is necessary to think critically about which predictors are important and not get bogged down by 'conventional' wisdom.

Literature Review

The National Basketball Association is a booming $5 billion per year industry and it grows ever larger. Teams are salivating at the chance to gain any type of competitive edge over their competition. Thanks in large part to the success of "Moneyball", or the use of advanced statistics in baseball, many basketball teams have turned to more sophisticated forms of analytics in order to help guide their decision making. As a result, the sports research market has flourished and continues to ask questions that appear simple but are in reality highly complex.

In Eric Nalisnick paper *Predicting Basketball Shout Outcomes*, he aims to do exactly what his title implies. The data set is fairly bare, as the only relevant variables are the player's position, the type of shot he attempted, and his location on the court in both the horizontal and vertical planes. Nalisnick chose to visualize the data and map each shot attempt on to an image of an NBA half-court and impose a classification boundary. Like us, he tested non-parametric methods such as kNN and logistic regression, in addition to trying parametric two-dimensional Gaussian models.

A notable difference in Nalisnick method is that his training set consists of shots from the 2006-2009 season with 2010 as his test season, whereas our data specified 5000 randomly chosen shots from Kobe's entire career. This may introduce more error into the model as team, player and league dynamics change from season to season. It is also worth noting that we only have data on one player who plays one position, whereas his data set draws on all players at all positions in the NBA. However, in his analysis it is noted that modeling all positions at once produces better results than going position-by-position. Because his data was processed to include binary variables, he decided like us that a nonparametric method would be best, namely logistic regression. A large part of the reason that both of our analyses chose to use logistic regression is because at its core, we are searching for a probability rather than attempting to cluster. In addition to that, we also have factor variables that need to be dummied out in order to compare them with our continuous variables. In his findings, he also notes that there is an upper bound to shot accuracy at around 90% due to the intrinsic error in attempting to shoot a basketball through a metal hoop many feet away. Nalisnick analysis is useful to us in that it takes a realistic approach to attempting to classify a shot and acknowledges the difficulty of the problem.

The paper from Lucey et al. *"Quality vs Quantity": Improved Shot Prediction in Soccer using Strategic Features from Spatiotemporal Data* asks a similar question and attempts to estimate the likelihood of scoring a goal given a certain opportunity. While this dataset is concerned with scoring chances in soccer, the basic principle is the same; given some positioning and type of shot, how likely is the player to score? In order to answer this question, the group partitioned shots into different match contexts such as penalties, corner kicks, etc., similar to how we categorized Kobe's shots like dunks and layups. However, their analysis differed from ours in that they had access to more data such as proximity and number of defenders, which was not available to us. This team used logistic regression to estimate the likelihood of each shot just as we did. Like us, they divided the data into training and testing sets to avoid over fitting.

Their analysis differed from ours in that it contained data on different teams and as a result they were able to analyze each team's efficiency in comparison to the others. Moreover, their analysis was also more fine grained than ours because of the additional data on every other player, so they could look at individual games and produce reasonable results. Where both of our analyses agree is on the fact that sports are random and outliers are likely to occur, such as a defender having a bad day or all of the shots for a particular team or player being successful.

References

Nalisnick, Eric. "Predicting Basketball Shot Outcomes." Nalisnick Blog. Wordpress, 23 Nov. 2014. Web. 01 June 2016. <https://enalisnick.wordpress.com/2014/11/24/predicting-basketball-shot-outcomes/>.

Lucey, Patrick, Alina Bialkowski, Matthew Montfort, Peter Carr, and Iain Matthews. “Quality vs Quantity”: Improved Shot Prediction in Soccer Using Strategic Features from Spatiotemporal Data (n.d.): n. pag.Disneyresearch.com. Disney, 28 Feb. 2015. Web. 1 June 2016. <http://www.disneyresearch.com/wp-content/uploads/Quality-vs-Quantity”-Improved-Shot-Prediction-in-Soccer-using-Strategic-Features-from-Spatiotemporal-Data-Paper.pdf>.

Appendix

```
# import data
library(data.table)
kobe = data.table(read.csv("kobe.csv"))
kobe = kobe[complete.cases(kobe),]

# randomly partition data into training set and test set
test.index = sample(seq_len(nrow(kobe)), size = floor(nrow(kobe)*0.1),replace = F)
train.index = setdiff(seq_len(nrow(kobe)),test.index)

shot.season=aggregate(shot_made_flag~season, kobe, length) # number of shots made per
season
plot(shot.season,xlab="Season",ylab="Number of attempted field goals", main="Frequency of
Attempts")
season.avg=aggregate(shot_made_flag~season,kobe,mean) # goal percentage per season
plot(season.avg,xlab="Season",ylab="Field goal percentage", main="Percentage of Shots
Made")

# transform data to type date
kobe$game_date = as.Date(kobe$game_date, "%Y-%m-%d")
# latitude and longitude are obviously useless
# game_id, game_event_id,team_id,team_name also useless
kobe[,lon:=NULL]
kobe[,lat:=NULL]
kobe[,game_id:=NULL]
kobe[,game_event_id:=NULL]
kobe[,team_id:=NULL]
kobe[,team_name:=NULL]

# combine minuetes remaining and seconds remaining
kobe[,seconds:=minutes_remaining*60+seconds_remaining,by=1:nrow(kobe)]
kobe[,minutes_remaining:=NULL]
kobe[,seconds_remaining:=NULL]

# replace matchup with binary variable "home"
kobe[,home:=1]
kobe[substr(matchup,5,5)=='@',home:=0]
kobe[,matchup:=NULL]

# some teams changed their names or locations
# we need combine old names and new names
kobe[opponent=="NOH",opponent:="NOP"]
kobe[opponent=="VAN",opponent:="MEM"]
kobe[opponent=="SEA",opponent:="OKC"]
kobe[opponent=="NJN",opponent:="BKN"]
library(rminer)
kobe$opponent = delevels(kobe$opponent, c("NOH","VAN","SEA","NJN"), label = NULL)

# library(corrplot)
```

```
# kobe.keep = kobe[,c("period","shot_distance","loc_x","loc_y","playoffs"),with=F]
# corr = corrplot(cor(kobe.keep))
# shot distance has a high correlation between loc_y
# but it doesn't effect our model very much

# check if we need action_type
action.glm = glm(data = kobe, shot_made_flag~action_type)
# summary(action.glm) output too long -> hide
combine.glm = glm(data = kobe,shot_made_flag~combined_shot_type)
# summary(combine.glm) output too long -> hide

# not all action type are statistically significant
# we will replace action_type with combined_type
# if specific action_type is infrequent compared to others

freqTable = data.table(action_types = levels(kobe$action_type),frequency =
as.vector(table(kobe$action_type)))
freqTable =  freqTable[frequency>=50]
freqTable
kobe[,type:=combined_shot_type,by=1:nrow(kobe)]
kobe[action_type %in% freqTable$action_types,type:=action_type]

# delete action_type and combined_type
kobe[,action_type:=NULL]
kobe[,combined_shot_type:=NULL]

# use chi-square test to see if these "distance/location variable" are independent
tbl1 = table(kobe$shot_type,kobe$shot_zone_area)
chi1 = chisq.test(tbl1,correct = F)
chi1
tbl2 = table(kobe$shot_type,kobe$shot_zone_basic)
chi2 = chisq.test(tbl2,correct = F)
chi2
tbl3 = table(kobe$shot_type,kobe$shot_zone_area)
chi3 = chisq.test(tbl3,correct = F)
chi3
# as expected they are highly dependent
# we will eventually drop them in the following procedure

# decision tree
library(rpart)
orig.tree = rpart(data = kobe,formula = shot_made_flag~.-shot_id,na.action =
NULL,control=rpart.control(minsplit=30, cp=0.001))
plot(orig.tree)
text(orig.tree)

# random forest
library(randomForest)
```

```
kobe.rf = randomForest(formula=as.factor(shot_made_flag)~.-shot_id,na.action=NULL,data =
kobe,ntree=500)
imptplot = varImpPlot(kobe.rf)
imptplot = as.data.table(imptplot,keep.rownames = T)
impt.sort = setorder(imptplot, cols = "MeanDecreaseGini")

col.keep = c(impt.sort$rn[7:15],"shot_made_flag","shot_id")
kobe.keep = subset(kobe,select = col.keep)
test = kobe.keep[test.index,]
train = kobe.keep[train.index,]

# redo random forest on training data
keep.rf = randomForest(formula=as.factor(shot_made_flag)~.-shot_id,na.action=NULL,data =
train,ntree=500)
imptplot = varImpPlot(keep.rf)

# random forrest prediction on training data
rand.train = predict(keep.rf,train,type="class")
rand.train.table = table(rand.train,train$shot_made_flag)
rf.train.error = (rand.train.table[3] + rand.train.table[2])/nrow(train)
rf.train.error

# random forrest prediction on test data
rand.test = predict(keep.rf,test,type="class")
rand.conti.table = table(rand.test,test$shot_made_flag)
rf.error.rate = (rand.conti.table[3] + rand.conti.table[2])/nrow(test)
rf.error.rate

glm.fit = glm(data = train, shot_made_flag~.-shot_id,family = binomial)
glm.fit

# logistic regresssion prediciton on training data
glm.probs.train = predict(glm.fit,train,type="response")
glm.pred.train = rep("Shot fail",nrow(train))
glm.pred.train[glm.probs.train>0.5]="Shot Made"
glm.conti.table.train = table(glm.pred.train,train$shot_made_flag)
glm.error.rate.train = (glm.conti.table.train[3] + glm.conti.table.train[2])/nrow(train)
glm.error.rate.train

# logistic regression prediction on testing data
glm.probs = predict(glm.fit,test,type="response")
glm.pred=rep("Shot fail",nrow(test))
glm.pred[glm.probs>0.5]="Shot Made"
glm.conti.table = table(glm.pred,test$shot_made_flag)
glm.error.rate = (glm.conti.table[3] + glm.conti.table[2])/nrow(test)
glm.error.rate
```