Using K-nearest Neighbors and K-means to Analyze Pitch Location and Pitch Type Data

Eric Gerber

Department of Statistics, Purdue University.
250 N. University Street, West Lafayette, IN 47907-2066, USA

May 7, 2015

Abstract

Effective pitching is key to winning games in the sport of baseball. Pitchers tend to have a wide selection of pitches to choose from in approaching the hitter, a "repertoire". What makes an effective use of this repertoire is something seasoned scouts and coaches have spent years understanding. Using the statistical methods of k-nearest neighbors and k-means, mixing and matching of the pitches and their location is analyzed using data from a single Major League Baseball game in which one of the pitchers threw a "perfect" game. We find that the two clustering algorithms tend to corroborate conventional strategy for pitch location.

Keywords: K-nearest Neighbours, K-means, Clustering, MLB, Pitch Location, pitchFx.

1. Introduction

The action of the sport of baseball is mainly centered on two focal players, a pitcher who throws a ball to a hitter who attempts to hit the ball and reach base. The hitter stands beside the home plate, the plane of which extends above to form a strike zone. A pitch that crosses through the strike zone is called a strike in favor of the pitcher, those outside are called balls and the hitter is under no obligation to swing at them. There are many different pitchers in the sport, and many have a variety of pitch types they specialize in throwing, such as fastballs or changeups. Conventional wisdom of pitching tactics call for faster pitches to be thrown higher in the zone and slower, more

movement oriented pitches to be thrown lower or outside. This project utilized k-nearest neighbors and k-means to classify pitches and define clusters by pitch type. Functions for accomplishing the k-nearest neighbor step were coded in R by hand, while the built-in k-means function in R[1] was utilized.

2. Description

Major League Baseball, the governing body of the highest level of professional baseball, makes available all pitch location (called pitchFx) data for every game. The data is collected by several cameras located in every stadium used in the League. While many variables are measured, for our purposes, only x-y coordinates of each pitch as they cross the plane of the plate, and the type of pitch thrown, are needed for this project. Two main data sets were used, both from the July 23, 2009 game between the Tampa Bay Rays and Chicago White Sox. The first, primary, data set included all 244 pitches thrown by all pitchers from both teams, while the second data set included the 116 pitches thrown by Mark Buehrle, the starting pitcher for the White Sox. This data set was chosen since Buehrle pitched the entirety of the game for the White Sox, and thus his pitches could be separated easily and also provide a relatively large sample from a single source. This data is also notable because Buehrle pitched a perfect game, in which not a single batter from the Rays managed to successfully reach base. Pitches from such a game would be expected to thus be more effective, considering successful pitch location by a pitcher can have a significant impact on their overall success in the game. There is a package in R, the pitchRx package[2], which makes collecting the data and organizing it for analyzing simple. The code for collection and preparation of the data is made available in a GitHub repository, with the link provided at the end of the paper.

From the overall game seven different pitch types were thrown, while Buehrle himself threw six different pitches. Below are the numerical breakdowns for type of pitch thrown for each dataset (Table 1), as well as the graphical representation of the pitches and their locations (Figure 1). The box in each graph is the designated strike zone, and the key for the pitches is as follows:

Key: (CH - Changeup, CU - Curve Ball, FC - Cut Fastball, FF - Four-seam Fastball, FT - Two-seam Fastball, SI - Sinker, SL - Slider)

Table 1: Pitch Selection

Pitch Type	CH	CU	FC	FF	FT	SI	SL
Overall Game	52	20	22	59	17	57	17
Mark Buehrle	34	16	10	42	13	0	1

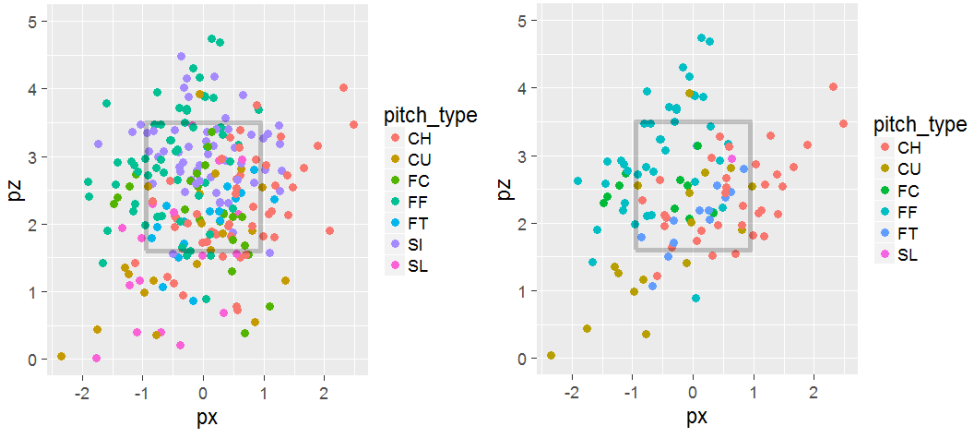

Figure 1: Comparison of overall (left) and Mark Buehrle (right) pitch location and type

While it is difficult to discern much of a pattern from the initial plots, it seems that, at least for Mark Buehrles pitches, there seems to be some evidence of clustering in terms of fastballs in the top left, curve balls in the bottom left, and changeups to the right. The area of the strike zone is not very large, so seeing any clustering with the naked eye would be promising.

3. Methods

Within the K-nearest neighbors function, the distance from each point to every other point was calculated using the Euclidean distance. Then, for any given k, the indices of the k-nearest neighbors were found and matched to the labels, the majority label of which was found and placed in an output vector, resulting in a vector with the pitch type classification for each pitch based on the k-nearest neighbors. The final function coded in R is presented below (Code 1), with get_knn the function that returns the indices of the k-nearest neighbors and maj_lab the function that matches the associated pitch type:

```
my_knn <- function(k, my_data){
  labs <- c()
  for(i in 1:nrow(my_data)){
    k_ind_vec <- get_knn(k, my_data[i,], my_data[-i,])
    labs <- rbind(labs,maj_lab(k_ind_vec))
  }
  return(labs)
}
```

Code 1: k-nearest neighbors function

While the function does as intended, there is the question as to what value k should take. In order to maximize the percentage of correct classifications based on k, each potential value of k was cycled through and the k which returned the highest number of correct classifications was used for both the overall and Buehrle datasets. The function used to choose the best k is below (Code 2) where the correct function finds the percent of correctly classified pitches for each potential k. For the overall dataset, the accuracy of the k-nearest neighbors function peaked at 38.11% using the 57 nearest neighbors for each point, while the Buehrle datasets provided 22.41% correct classification with k=7. For the overall dataset, only the three most prevalent pitch types (levels) were preserved, while all levels were persevered in the classification of the Buehrle data. The plots of these classifications are presented in the results section (Figure 3).

```
bestK <- function(my_data, tru_labels){
  acc <- c()
  for(i in 1:(length(tru_labels)-1)){
    acc[i] <- correct(i, my_data, tru_labels)
  }
  return(list(which.max(acc[]), max(acc), acc))
}
```

Code 2: optimal choice of k function

Lastly, for a nonparametric approach to clustering and contrast to k-nearest neighbors, k-means was used to find clusters associated with the pitch types. The average locations of all types of pitches was calculated and served as the initialization points for the clusters within the kmeans function. This guaranteed that there would end up being the same number of clusters as pitch

types, which is not a guarantee when using k-nearest neighbors. The kmeans function ended up providing cluster classifications that were slightly worse (27.87% correctly classified) for the overall game pitch data, yet significantly better (44.83% correctly classified) for Mark Buehrles pitches. The major results are discussed in the next section.

4. Results

The k-nearest neighbors classification did not provide a very clear delineation of pitch type by location, especially since for the overall game k-nearest neighbors only produced pitch classifications for the three most prevalent pitch types. However, both k-nearest neighbors and k-means seem to show classification of faster pitches higher in the zone and slower pitches (especially changeups and curve balls) lower in the zone. For the overall data, sinkers tended to be in the upper half of the strike zone, which seems counter-intuitive. However, Buehrle (the winner of the game) does not throw a sinker, thus the sinker was thrown by the pitcher of the losing team and potentially their ill-advised location is related to that fact.

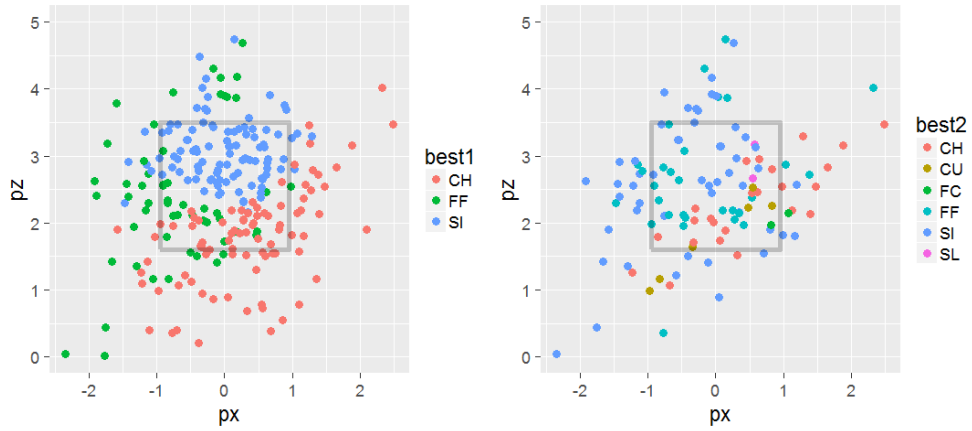

Figure 3: Comparison of overall (left) and Mark Buehrle (right) k-nearest neighbors pitch type classification

While the k-nearest neighbors algorithm did not manage to recover all the pitch type classifications for the overall dataset, it did for the Buehrle data

set, and did so correctly 22.41% of the time. This would seem to indicate that there is some intrinsic relationship between location and type of pitch, since a random classification of Buehrle's pitches should have resulted in correct classification at a rate of approximately 16.67%.

K-means provided a much clearer delineation of pitch type based on location. The two sets of plots below (Figures 4, 5) provide two plots for each the overall game and Buehrles pitches:

1. A plot with the points assigned the color of their cluster owing to the kmeans function, with the initial means of the different pitch types represented by colored regions, and
2. A plot with the original pitches and their types, with the regions of the cluster assignments found via the kmeans function.

While the number of correct classifications were lower for the overall game with the kmeans function that with k-nearest neighbors, the k-means algorithm is designed to maintain the correct number of clusters, and correctly classified pitches at a better than random clip. K-means took a rather interspersed collection of pitches and found that curve balls and sliders formed lower clusters, sinkers formed the highest, and changeups, two-seam fastballs, four-seam fastballs, and cut-fastballs all hovered around and to the sides of the strike zone (Figure 4).

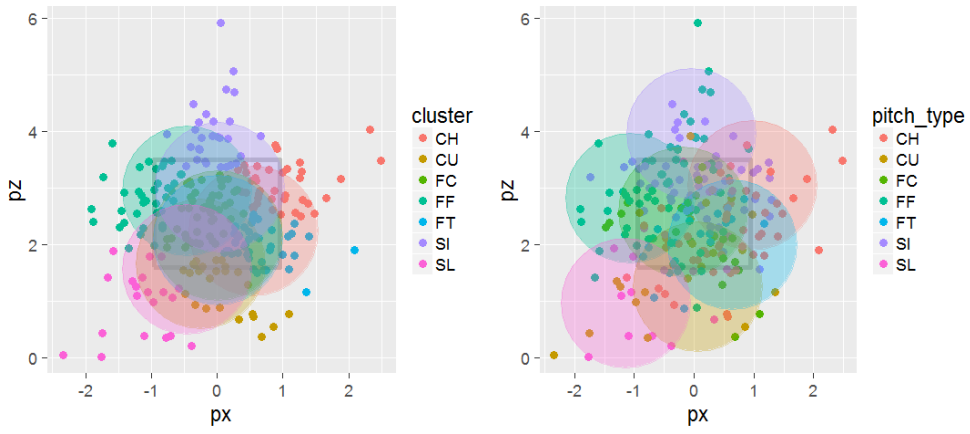

Figure 4: K-means cluster assignments for pitch type in overall game

Buehrles pitches, using k-means, were correctly identified an impressive nearly 45 percent of the time, and showed that, much as conventional wisdom holds, success can be found by keeping slow curve balls low and out of the zone, while four-seam fastballs stay high. In fact, the cluster locations for each pitch except the slider are almost precisely where a left-handed pitcher (such as Buehrle) would want to aim their pitches. Curve balls in the dirt, change-ups low and away to right-handed hitters, cutters running off the plate to the left, four-seam fastballs high, and two-seam fastballs low and away to left-handed hitters (Figure 5).

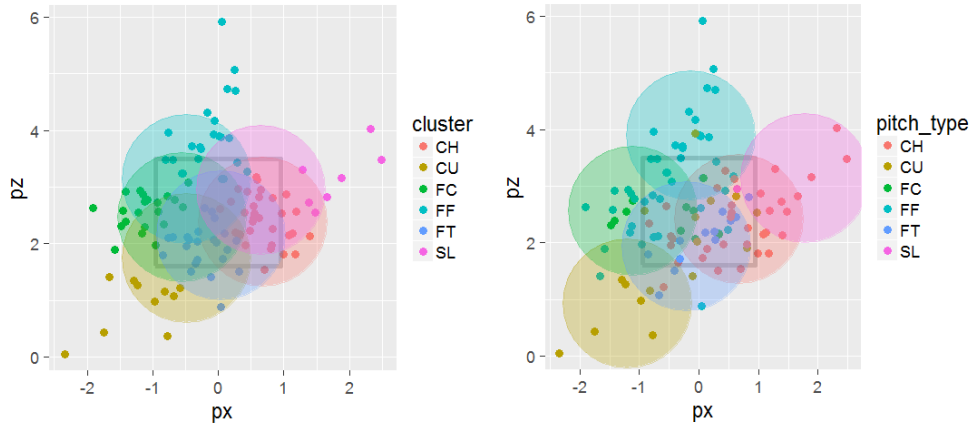

Figure 5: K-means cluster assignments for pitch type of only Mark Buehrle's pitches

If we look back at the initial summary table, we find that Buehrle threw only one slider during the game, which might explain why a game in which he was so successful resulted in a cluster for sliders opposed to traditional strategy.

5. Conclusions

Based on the results it does seem that pitch location and type of pitch thrown are related, and there appears to be evidence that success as a pitcher can be in some sense inferred from being able to locate certain types of pitches in certain locations in the strike zone. In this case, conventional baseball wisdom, based on sound logic, is at least somewhat validated by analysis. Apart from simple classification, K-nearest neighbors and k-means provide an interesting avenue for analyzing a pitchers tendencies. Ideally, a more concrete

connection between specific pitch location and success could be made my somehow incorporating outcome of the pitch, instead of simply using data from a game in which a pitcher was successful and attributing each pitchs location that success. As a start, Buehrles perfect game does show some trends in what pitch selection and location can tell us about successful pitching.

6. Code

All code is available via a GitHub repository:
https://github.com/gerber19/knnkmeanspitch/.

7. Acknowledgments

While not cited, the inspiration for this project and the knowledge of how to retrieve the necessary data came from *Analyzing Baseball Data with R*, by Max Marchi and Jim Albert, part of The R Series by Chapman & Hall/CRC. I would like to thank Professor Vinayak Rao for allowing us to choose our own projects for his Introduction to Computing for Statisticians course, and the TA Mohit Dayal for helping with some of the intricacies of the kmeans function in R on this project in particular.

8. References

[1] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2014.

[2] Carson Sievert. *pitchRx: Tools for Harnessing MLBAM Gameday data and Visualizing PITCHf/x*, 2014. R package version 1.6.