Identifying Injury-Inducing Factors in Baseball Pitchers

Jae Jang Raejoon Jung Prafull Sharma

Abstract

This paper aims to identify which metrics in baseball are meaningful in predicting whether a pitcher will get injured before a particular season. To do so, we define a cost function and build a framework that takes into account the asymmetric reward and cost associated with correctly (or incorrectly) predicting injuries. We then vary the sets of features on various machine learning techniques (i.e., SVM and Linear Regression) to test each model within our framework and compute a utility score for each model. Finally we use this utility score in conjunction with other standard feature filtering techniques such as PCA and mutual information analysis to determine which features provide the most useful information regarding whether a player will get injured in the context of a season.

1 Introduction

The integration of technology in professional sports and recent advances in wearable technology has led to an exponential increase in the amount of data available for sports analytics. This is especially true for baseball, where many of its performance metrics are readily quantifiable and are recorded consistently across different games and teams. With the abundance of data, it becomes increasingly challenging to pinpoint which data is meaningful and to determine how beset to use the data. In this paper, we address a problem that plagues not only individual players, but major league baseball teams year in and year out: pitcher injuries. Having a healthy pitcher for a starting season crucially impacts the outcome of a season, and teams and managers consequently invest millions of dollars into acquiring and identifying pitchers who can stay healthy for a full season. With this context in mind, we focus specifically on identifying the features that are useful for predicting injuries before a season. For this task, choosing features based on correlation or mutual information analysis alone has a few shortcomings as these techniques do not take into account the relationship of the predictor variables to the response variables. In our approach, we first define a cost function that reflects the cost and reward for predicting injuries and build a framework that provides a meaningful measure of how well a model performs in predicting an injury based on this cost function. Finally, we support the results by verifying that it is consistent with other feature filtering techniques and provide an intuition for each of the selected features.

2 Related Work

The prediction of injury in baseball has been a topic of interest for a long time due to the financial implications of being able to do so successfully. Much of the past research and papers make the conclusion that the best indicator of future injury is whether a player has suffered injury or undergone any type of arm surgery in the past. [9] As examples of such past work, we point to two articles that employ well-known machine learning techniques, forward search and Random Forests. Carleton [7] uses a forward stepwise model to rank the usefulness of features as indicators of future injuries and concludes that the one feature that is consistently meaningful in predicting an injury is past injury occurrence. In [4], the study uses random forests as a means of injury prediction and reaches a conclusion similar to that of the first paper, that past injury is the best indicator of future injury.

We argue, however, that past injury is a signal that general managers and team owners are already taking into account; baseball teams are reluctant to trade for players with past injuries, and the stock of a pitcher plummets after an incidence of injury or surgery. [5] We attempt to answer a more challenging problem in our paper; the problem of predicting injuries for players without past injuries.

3 Dataset and Features

We combined data from two databases to create our dataset. From [1], we obtained player statistics on all MLB pitchers for the past five seasons, with each year containing 33 features for each player. We then used injury data from [2] to label each sample in the first dataset. The injury data provides the date and name of every Major League player to undergo Tommy John surgery during their career since 1986, and in the labeling process, we matched each sample by name, team, and year to avoid duplicate name collisions. We were able to preserve all our samples using this method as there were no instances of a team ever having two pitchers with the same name in our dataset. We elected to use the data in two ways. In one approach, we treated data from different years as different input features and in another, we used average and cumulative statistic for each player as input.

When treating data from different years as different input features, We elected to use a mapping from "Player Name" to "Career Statistics". Because every player contains data from different seasons (not always contiguous), we packaged the set of statistics for different years into another mapping from "Season(year X)" to "Statistics for year X." The following steps were also taken to merge the datasets and filter certain features, with a brief rationale provided for each.

1. *Remove players who moved teams during a season via trades/transfers*
   These players show up multiple times in a year and because some features are sums, while others are averages over numbers of games, and others yet are averages over innings or balls pitched, we decided 59 such instances from our dataset. None of our positive samples fell under this case.

2. *Remove seasons after a player got injured*
   Our goal is to make predictions before a player gets injured, and we thus elected to remove data for seasons an injured player played after coming back from his most recent surgery since the player would then be classified as having a past injury occurrence. Also, because some players had surgery multiple times, we also explored how the result differed if we removed seasons after a players *first* surgery, or did not remove any seasons at all. Our initial rationale behind electing to remove only seasons after the *final* surgery was that the means of positive and negative samples were most different in this case and we provide a more formal justification in the context of data noise in the results section.

3. *Features removed*
   Features like high school, college of players were removed as many players (especially foreign players) has these fields missing in this data and very few of the players seemed to share same instances of these two features.

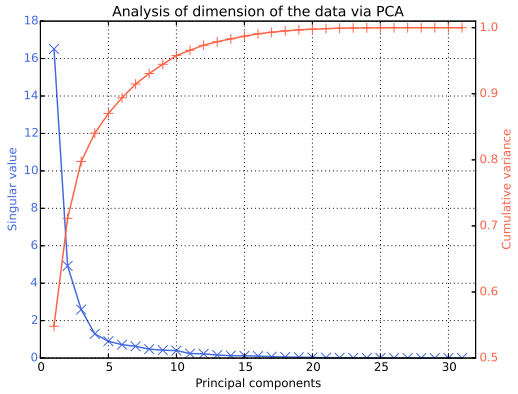

Figure 1: Dimensionality of the data

After all the features were removed, our dataset consists of 1095 samples (players) and 166 positives samples. We make particular note of the asymmetry in the data with positive samples comprising only about 7% to 15% of the data depending on the year, a fact that will become much more relevant in the context of our testing framework.

Finally, we use two scaling methods on the data throughout our process, normalization and standardization; we either normalize the columns of each feature to a [0, 1] scale or standardize each feature set to have zero mean and unit variance. Additionally, we performed a Principal Component Analysis on the standardized data to get an initial picture of the variance and covariance between the features (Figure 1).

One important thing to note is that despite having over 30 features, less than half (15) principal components account for 99% of the variance in the dataset. This is largely due to the fact that many of our features were calculated from one another, there was a high correlation between the features. As a concrete example, we point to WHIP, which is calculated as (Walks + Hits) / Innnigs Pitched, all three of which are also features in our dataset.

4 Methods

As a baseline approach, we ran k-fold using the entire dataset using GDA, Linear Regression, and SVM. The results were consistent in all the attempts, with GDA underperforming in nearly every case. Regardless of the model, we achieved a deceptively high prediction accuracy of 91-93% with each test. We describe the result

as deceptive because upon examination, we found that our models were always predicting a negative label for all input instances, and the test accuracy corresponded exactly to the proportion of negative samples in the test data (91-93%). This problem is magnified in the K-folds approach as some of the training chunks sometimes receive very few or even zero positive samples, leading to a model that predicts negative for any given input.

As an initial attempt at curing this result, we mapped the features to higher dimensions and use SVM with various kernel function to see if it resulted in clearer separation. Even when using a polynomial of the second order, however, we achieved a training accuaracy of nearly 100%, but the test error fell below 75%. This confirmed that increasing the number of features, especially given the high correlation between our predictor variables, was not the correct path to explore and that we needed a different approach to address the asymmetry of of the data. We intergrated two ideas into our model: down-sampling the majority and defining a cost function to offset the level of asymmetry.

Cost Function

By using test error as our only measure of accuracy, we were failing to capture one very important characteristic about predicting injuries: the cost of labeling an injured player as healthy was much higher than the cost of labeling a healthy player as injured. To provide a little more intuition on this notion, it makes sense from the economic perspective that the cost of drafting/acquiring an injured player is much higher than the (opportunity) cost of not drafting a healthy player. Thus, the total test error alone, wasnt the most appropriate measure of how accurate our predictions in the context of baseball, and we defined our own cost function in order to capture this difference in the cost of misclassification. We adopt a method discussed in Bach et al.s work on handling asymmetry of data with SVM for our cost function. [6] We formally define the cost function for SVM as follows:

$$C(c_+, c_-, \hat{y}) = \frac{c_+}{n} \sum_{i=1}^{n} \mathbb{1}\{\hat{y}_i = -1, y_i = 1\} + \frac{c_-}{n} \sum_{i=1}^{n} \mathbb{1}\{\hat{y}_i = 1, y_i = -1\} \quad (1)$$

$$U(c_+, c_-, \hat{y}) = -C(c_+, c_-, \hat{y}) \quad (2)$$

where $\hat{y}_i = h(x_i, w, b)$. c_+ and c_- are costs of false negative and false positive errors, respectively.

$$\frac{\text{Estimated cost of drafting an injured player}}{\text{Estimated cost of not drafting a healthy player}} \approx \frac{c_+}{c_-} = \frac{\text{Number of negative samples}}{\text{Number of positive samples}} \quad (3)$$

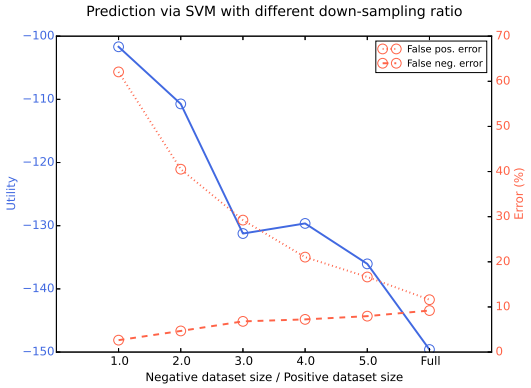

Figure 2: Prediction via SVM with different down-sampling ratio

Majority Down-sampling

Another idea we implemented as part of the training process was to down sample the majority to a factor of the positives samples. In addition to the full training set, we used 5 different negative/positive sample ratio to train each algorithm. A simple implementation of varying the ratio, however, led to very arbitrary result depending on the run. Upon investigation, we found that the reason for this was that because of the small ratio of positives samples, in some instances of random down-sampling, the training set contained little to no positive samples, and the hypotheses in these cases led to predictions with an overwhelming number of false negatives. In one approach, we kept the number of positive samples fixed for each k-fold instance, but this introduced the problem of our training set and testing set being too similar. To remedy this issue, the final approach we adopted was to (1) separate our positive samples from the negative samples, (2) select the 90% (training) and 10% (testing) chunks from each respective set, (3) combine and shuffle the two randomly selected chunks to ensure that every training set was guaranteed some positive samples.

5 Results and Discussion

Of the many variables with which we experimented, we elected to specifically display the test results for the two more interesting methods we implemented into our framework: majority down-sampling and positive/negative cost ratio. The test results from different ratios (using SVM with a Gaussian Kernel) are displayed in figure 2 As shown in the figure 2 above, the overall test error, especially the false positive error, is highest when using a down-sample ratio of 1:1 ratio, but the high test error is also accompanied by a high utility. In fact the

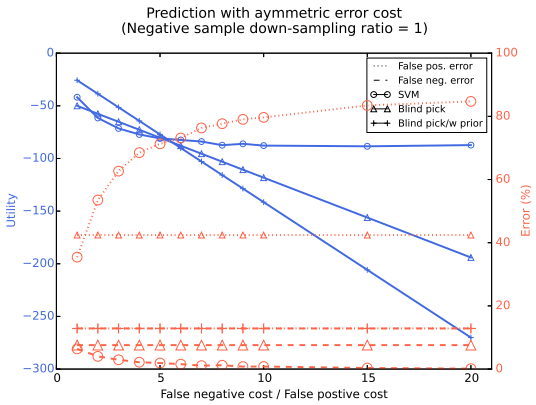

Figure 3: Prediction with asymmetric error cost

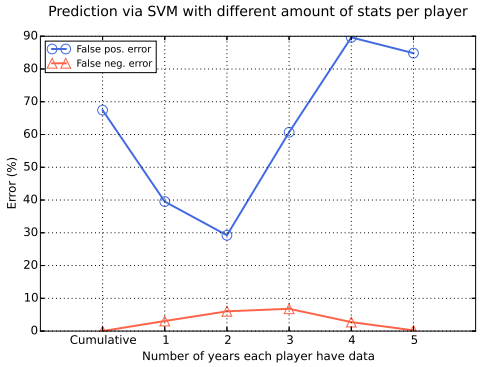

Figure 4: Prediction with different amount of stats per player

utility decreases continually as we increase the down-sample ratio (decrease the asymmetry of data in the training set). Intuitively, this is consistent with exactly what we were trying to integrate into our model. The cost of failing to identify an injury is much higher than the cost of mis-labeling a healthy player as "injured", and this is illustrated in the figure. The highest utility is achieved at a level when the instances of false negatives are low even at the cost of having relatively high number of false positives. Figure 3 plots the results over our second variable of interest, the penalty ratio (Cost of False Negative / Cost of False Positive) used in the SVM cost function with the down-sample ratio fixed at 1:1. We compare our model against two different baseline models, one using a blind random pick (50/50 chance) and one using a random pick with knowledge of the prior. To elaborate, the "prior" baseline model uses its prior knowledge to make predictions. For example, if the prior is the 90% of the players are healthy then it predicts that the player will be healthy with a probability of 0.9. Figure 3 plots the corresponding utilities of these two baseline models for every penalty ratio. When the penalty ratio is less than 6, SVM performs slightly worse than the other two models but achieves a higher utility for all other ratios.

Bach et al.s paper recommends a penalty ratio corresponding to (num majority samples / num minority samples) to offset the asymmetry of the data, but the reason we test our model using different penalty ratios was that the true ratio of the cost of false negative to the cost of false positives in baseball is unknown and we ideally wanted our model to as robust as possible and outperform the baseline models for different penalty ratios. We extend the test cases all the way to a penalty ratio of 20 based on the intuition that the cost of investing millions in a soon-to-be-injured player is much higher than maintaining the incumbent player at the current position (presumably much higher than the empirical ratio of 10 based purely on the ratio of pos/neg samples).

We also highlight one additional variable of interest introduced in the data section. As mentioned briefly as part of data processing, we viewed the number of years prior to an injury as an important factor in training our model, especially given the fact that our dataset was obtained with years as different cross sections from the database. We thus varied the number of years prior to an injury to use with several models as we made changes to the cost function and down-sample ratio. Figure 4 illustrates the % errors using different number of years. The lowest false positive error is achieved using data from the most recent two years before a particular season while the count of false negatives is relatively constant for all down-sampling ratios. One reasonable interpretation might be that too few years leads to a lack of data to accurately predict the results while using too many years (where statistics from each year mean additional feature variables) results mostly in additional noise while not adding much more meaningful information.

6 Conclusion/Future Work

Using the framework that we developed, we applied a forward-search process on all our features and were able to rank the features in the order selected by the process. We list the 10 highest ranked features in figure 5, along with the mutual information for each variable of interest. When testing using only these 10 features versus the entire set of 33 features, we achieve a utility of -25.56 versus -23.32. We commit the remainder of the discussion to provide an interpretation for each of the selected features.

The above features can be classified into three major categories: pitch velocity, workload per season, and workload per outing. The first category is best described

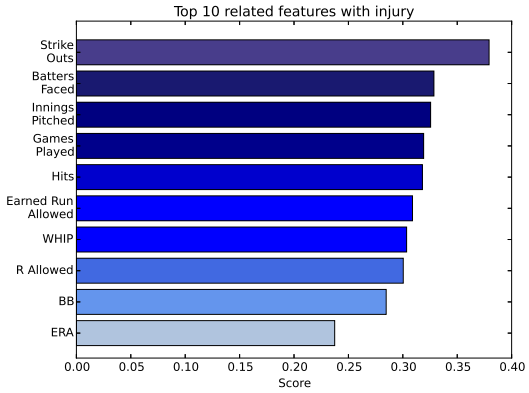

Figure 5: Top 10 related features with Tommy John Surgery

as representing the velocity of the pitch. Strike Outs, WHIP, R allowed, and ERA are all measures that are associated with a pitcher with throwing velocity. A pitcher with a strong fastball in his repertoire is more likely to have a higher strikeout, lower WHIP, Runs Allowed, Hits, and ERA but also more prone to injuries than pitchers who rely on finesse and change-ups. [10] The second category is the workload per season, or pitching volume, of the pitcher as measured in innings the pitcher has to handle per season. Batters Faced and Innings Pitched are traditional measures of the workload volume a pitcher in a season and pitchers with high values in these two features become susceptible to injuries from overuse. [8] The third category represents the length of time a pitcher is out on the field in one outing. This quantity is best measured by hits, BB (walks) as pitchers with a high number in these two categories usually stay on the plate longer per inning than the average pitcher.

As possible areas of further research, we would like to explore integrating detailed data about pitcher physiology and mechanics. Examples of such features include their throwing motion, the height of ball release, average velocity of pitches. [3]

Another area of future research involves not relying on the asymmetry of data to define the relative penalty costs but use a ratio that is reflective of the cost and reward associated with drafting or not drafting a player in the real world. This might be done by comparing the financial cost of drafting a soon-to-be-injured player to the opportunity cost of foregoing drafting a healthy player. Using salary information and modern baseball metrics such as WAR (Wins above Replacement) may provide a measure of such an opportunity cost.

7 References/Bibliography

[1] http://www.baseball-reference.com.

[2] Disabled list data. http://www.baseballheatmaps.com/disabled-list-data.

[3] Young arms and curveballs: The real story behind it all, 2012. http://www.drivelinebaseball.com/.

[4] Predicting injury status, 2014. http://makenolittleplans.net/predicting-injury-status/.

[5] Injuries impact newly expanded top draft prosepects list, 2015. http://m.mlb.com/news/article/120823800/injuries-impact-landscape-of-top-100-draft-prospects-list.

[6] BACH, F. R., HECKERMAN, D., AND HORVITZ, E. Considering cost asymmetry in learning classifiers. *The Journal of Machine Learning Research 7* (2006), 1713–1741.

[7] CARLETON, R. A. Baseball therapy: What really predicts pitcher injuries, 2013. http://www.baseballprospectus.com/article.php?articleid=19653.

[8] PARKS, E. D., AND RAY, T. R. Prevention of overuse injuries in young baseball pitchers. *Sports Health: A Multidisciplinary Approach 1*, 6 (2009), 514–517.

[9] ZIMMERMAN, J. 2015 starting pitcher dl projections, 2014. http://www.fangraphs.com/fantasy/2015-starting-pitcher-dl-projections.

[10] ZIMMERMAN, J. Velocity's relationship with pitcher arm injuries, 2015. http://www.hardballtimes.com/velocitys-relationship-with-pitcher-arm-injuries/.