A Data-driven Method for In-game Decision Making in MLB

When to Pull a Starting Pitcher

Gartheeban Ganeshapillai
Massachusetts Institute of Technology
77 Massachusetts Avenue,
Cambridge, MA 02139
garthee@mit.edu

John Guttag
Massachusetts Institute of Technology
77 Massachusetts Avenue,
Cambridge, MA 02139
guttag@mit.edu

ABSTRACT

Professional sports is a roughly $500 billion dollar industry that is increasingly data-driven. In this paper we show how machine learning can be applied to generate a model that could lead to better on-field decisions by managers of professional baseball teams. Specifically we show how to use regularized linear regression to learn pitcher-specific predictive models that can be used to help decide when a starting pitcher should be replaced. A key step in the process is our method of converting categorical variables (e.g., the venue in which a game is played) into continuous variables suitable for the regression. Another key step is dealing with situations in which there is an insufficient amount of data to compute measures such as the effectiveness of a pitcher against specific batters.

For each season we trained on the first 80% of the games, and tested on the rest. The results suggest that using our model could have led to better decisions than those made by major league managers. Applying our model would have led to a different decision 48% of the time. For those games in which a manager left a pitcher in that our model would have removed, the pitcher ended up performing poorly 60% of the time.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Applications and Expert Systems; G.3 [**Probability and Statistics**]: *Multivariate statistics, Time series analysis*

General Terms

Machine Learning Applications

Keywords

MLB, Predictive Modeling

1. INTRODUCTION

Perhaps the most important in-game decision a baseball manager makes is when to relieve a starting pitcher [2]. Today, managers rely on various heuristics (e.g., pitch count) to decide when a starting pitcher should be relieved [2, 17]. In this paper, we derive from data a model that can be used to assist in making these decisions in a principled way.

We pose the problem as classification problem of predicting whether a starting pitcher would give up at least one run if allowed to start the next inning. (This formulation seems appropriate late in close games, but is probably not the right formulation early in a game or during a blowout.) Our method uses information about the current at-bat, game situation, and historical data.

First, we reformulate our problem into a regression problem using Pitcher's Total Bases (PTB) [9] instead of runs allowed as the dependent variable because PTB is generally accepted as a better indicator of pitcher effectiveness. We solve the regression problem using regularized least squares to produce a pitcher-specific predictor for the expected PTB in the next inning. The main technical interest of this work lies in our approach to constructing the variables used in the regularized regression.

- Many of the important predictors that provide contextual information such as the next batter, the venue in which the game is played, etc., are categorical variables. We convert these categorical variables into continuous variables by deriving associations between these variables and various statistics in the historical data. For example, the batting team is represented by prior statistics such as the average number of hits against the current pitcher. We call these statistical associations *priors*.

- In those cases where the amount of data available to derive an association is small, we shrink the prior using global averages to prevent overfitting.

We evaluate our method on the MLB 2006-2010 data set from STATS Inc.[1], which contains a record of each pitch thrown in MLB games in both the regular and post seasons. We train our model using the first 80% of the games of each season and test it on the last 20%.

We evaluate our model relative to what actually occurred in the games in the test set. First, we learned a model that closely reflects the actual decisions made by MLB managers.

---

[1] http://www.stats.com/baseball.asp

It is well known that managers rely heavily on pitch count and the opposing team's scoring to determine when to relieve the starting pitcher [2, 17, 12]. This *manager model* learns pitcher-specific parameters that fit these two variables against the manager's decisions on the training data.

For the 76 pitchers who pitched at least 500 pitches in each year, our manager model accurately models the manager's decision in $95\%(\pm1.2\%)$ of the innings, i.e., it is a reasonably accurate model of what major league managers actually do. When the manager model is used to predict whether or not a run will be scored, it correctly predicts $75\%(\pm4.6\%)$ of the innings. In contrast, our model makes a correct prediction for $81\%(\pm4.9\%)$ of the innings. Our model also outperforms the manager model in F-score (0.41 vs 0.26) and odds ratio (3.2 vs 1.2).

Second, we demonstrate that applying our model would have led to a different decision 48% of the time after the fourth inning in close games. For those close game situations in which a manager actually left a pitcher in that the model would have removed, 60% of the time the pitcher surrendered a run. Unfortunately, it is impossible to say what would have happened in those situations where a manager removed a pitcher that the model would have allowed to continue, since we don't know whether or not the removed pitcher would have given up a run had he not been removed.

In Section 2 we present some background on baseball and related work. In Section 3 we describe our method. In Section 4 we discuss the measures of performance used to evaluate our method, and present the results of a series of tests in which comparisons are made using several performance measures. Section 5 discusses the real world applications of our method. In Section 6 we discuss the limitations and outline future work. Section 7 provides a conclusion.

2. RELATED WORK

Baseball was one of the first sports to attract statistical analysis. As a sport, it has all the necessary characteristics for performing a quantitative analysis: a rich dataset with detailed records for several decades, ordered game progression that allows activity to be nicely segmented, and a reasonable amount of independence between events. This has led to the development of many statistical methods for assessing individual baseball players over the years [14, 1, 11, 3].

Bill James used statistical data in the 1980s to analyze why teams win and lose. He termed his approach sabermetrics. In an effort to quantify the contribution of players to wins and losses he invented many statistical measures such as runs created, Component ERA, and similarity scores[16].

Baseball Prospectus, published by Gary Huckabay from 1996, uses a system called *Vladimir* to predict a player's performance in the next season using the context of the player's statistics such as the venue in which the game is played. It projects how the performance evolves as a player ages [8, 15].

In 2003, Nate Silver published a state-of-the-art sabermetric system for forecasting MLB player performance, termed PECOTA. It is a nearest neighbor based similarity metric that searches through thousands of players to find the ones with similar profiles. It computes the odds of a draft player having a successful future[14]. It extended the method to compute the statistical similarity between any two major-league players published in Baseball Abstract by Bill James

in 1986 [10]. Since then there have been many commercial systems developed to model the progression of players[1, 11, 3, 13, 15].

Other than [6], which described a method for predicting the next type of pitch, almost all of the existing baseball statistical methods address high level problems that span multiple games: projecting a player's performance over the years, evaluating a player's contributions to the wins and losses, and optimizing a team budget. In contrast, we present a machine learning method to assist decision making during a game. In particular we present a method for producing predictive models that can be used to help decide whether a pitcher should be replaced at various points in the game.

Pitchers are divided into two categories, starters (the first pitcher for each team) and relievers (all subsequent pitchers). Starters rarely pitch the complete game, and deciding when to replace the starter is often the most important in-game decision that a manager has to make. Early in the game the decision is based upon many factors, including the effectiveness of the starter and the managers desire to conserve the bullpen. Later in the game, the decision is based primarily on the manager's estimate of the relative expected effectiveness of the starter and the pitcher who would replace him. Our work is designed to provide useful information in the second half of close games.

3. METHOD

We first build a pitcher-specific predictor for the expected PTB in the next inning. We then use this prediction to build a classifier that predcits whether a run will be given in the next inning.

3.1 Features

The independent variable x of our model incorporates the following groups of information available to a manager at the time a decision has to be made:

- Current game statistics: team scores, outs, inning number, and pitch count,

- Averages of all previous innings in the season: strike and ball count, base codes (bases advanced), home runs, hits in play, walks (intentional walk discounted), steals, strike outs, and the distributions of pitch velocity and zones,

- Next three batters in the opposing line-up, and

- Venue (home team), batting team, inning, and defense configuration

Table 1 lists our feature vector.

3.1.1 Categorical Variables

By construction, our feature vector contains only continuous variables. However, many variables such as batter, batting team, and home team are naturally categorical. We use prior statistics to transform these variables into continuous variables. For example, the batting team is represented by prior statistics of how many times the pitcher faced the team, the runs given, and number of hits. We compile pitcher-home team, pitcher-batting team, pitcher-inning, and pitcher-defense configuration priors. For each of the next three batters in the line-up, we compile the averages of pitcher-batter priors and batter priors (independent

of pitcher). These priors are represented by how many times they faced each other, Component ERA (ERC) for the pair, slugging percentage (SLG) for the pair, and number of hits and runs between them.

Our predictors incorporate at-bat information, game statistics, and the context of the game statistics. For instance, the pitcher-venue prior implicitly captures information about the dimensions of the stadium where the game is played. Such information can provide critical predictive information for some pitchers and batters. For example, in 2012, the Boston Red Sox pitcher Felix Doubront's ERA was almost a full run higher in his home stadium (which has a short left field wall) than elsewhere.

These baseball statistics attempt to quantify certain characteristics and skills of the players that are not captured by runs alone. SLG measures the power of a hitter, ERA discounts luck by excluding non-earned runs, and component ERA includes partial runs, i.e., hits and walks.

3.1.2 Shrinkage

Sometimes, these priors may be unreliable because of small support. A particular pitcher might not have faced a particular batter, or thrown only a few pitches to that batter. In such cases, the prior statistics may not be meaningful. We improve the utility of values with low supports by shrinking them towards the global average [4]. For example, global average of pitcher-batter prior (e.g., SLG) is the pitcher prior against all the batters.

Suppose a random variable $s \sim N(\mu, \sigma^2)$ is observed via $\hat{s}|s \sim N(s, \sigma_s^2)$. Under the Bayesian interpretation, posterior mean s for observation $\hat{s}$ is given by

$$E(s|\hat{s}) = \frac{\sigma^2.\hat{s} + \sigma_s^2.\mu}{\sigma^2 + \sigma_s^2} \qquad (1)$$

For the pitcher-variable prior $\hat{s}$, global average p, support n, and some constant β, by making the assumption that σ_s^2 is proportional to $1/n$, we derive the shrunk pitcher-variable prior s

$$s = \frac{n.\hat{s} + \beta.p}{n + \beta} \qquad (2)$$

3.1.3 Luck and Randomness

When choosing the predictors, we try to minimize the influence of luck, and focus on variables that are less erratic and capture the root causes. For instance, we slugging percentage to capture the skill of batsman better than RBI. Similarly, strikeouts, walks, and steals yield predictive information about the skills of the pitching team. Also, when using balls and walks count, we discount intentional balls and intentional walks.

The actual number of runs fails to take into account luck and near misses. In baseball near term outcomes are often dominated by randomness. In order to take that into account, we use Pitcher's Total Bases (PTB) (Equation 3) in place of runs as the dependent variable. PTB is a factor in Component ERA, a baseball statistic invented by Bill James [3,9].

$$\begin{aligned} PTB = &0.89 \times (1.255 \times (H - HR) + 4 \times HR) \\ &+ 0.56 \times (BB + HBP - IBB) \end{aligned} \qquad (3)$$

Here, H is hit, HR is home run, BB is walk, HBP is hit by pitch, and IBB is intentional walk.

Table 1: Feature vector

	Current game statistics:
1	Batting team score
2	Pitching team score
3-5	Outs, Inning number and pitch count
	Previous inning Statistics (averages):
6-7	Strikes, balls (intentional balls discounted)
8-10	Bases advanced, home runs, hits in play
11	Walks (intentional walks discounted)
12-13	Steals, strike outs
14-21	Pitch Velocity & Pitch Zone (binned)
	Priors:
22-26	Pitcher-Inning (Count, SLG, ERC, Runs, Hits)
27-31	Pitcher-Batting team (Count, SLG, ERC, Runs, Hits)
32-36	Pitcher-Venue (Count, SLG, ERC, Runs, Hits)
37-41	Pitcher-Defense (Count, SLG, ERC, Runs, Hits)
42-56	Pitcher-Batter (1-3) (Count, SLG, ERC, Runs, Hits)
57-68	Batter (1-3) (SLG, ERC, Runs, Hits)

3.2 Problem Formulation

First, we formulate our problem as a regression problem, and solve it using regularized least squares [7]. For pitcher j, our training data is a set of n_j points of the form $\{(x_i^j, r_i^j)|x_i^j \in \Re^p, r_i^j \in \Re\}_{i=1}^{n_j}$, where r_i^j is PTB and x is feature vector (Table 1).

$$\min_{\hat{w}_j} \frac{1}{n^j} \sum_{i=1}^{n_j} (r_i^j - x_i^j.w_j^T)^2 + \lambda|w_j|^2 \qquad (4)$$

We next move to the binary problem of predicting whether there will be a run given in the next inning. We do this by binarizing the outputs of the regression r_i^j, and finding a pitcher specific cut-off $\hat{b}_j$ on the estimated PTB that maximizes the prediction accuracy on the validation set.

$$\max_{\hat{b}_j} \sum_{i=1}^{n_j} y_i(x_i^j.\hat{w}_j^T - b_j)^2 \qquad (5)$$

Prediction outputs are given by $sign(x_i^j.\hat{w}_j^T - \hat{b}_j)$.

3.2.1 Multi-task Learning

Our model is pitcher-specific, and we learn feature weights independently for each pitcher. However, we want to take advantage of the intuition that feature weights should have some relationship across pitchers (e.g., Fenway Park presents challenges for most left handed pitchers). Therefore, we rewrite Equation 4 with a multi-task learning formulation [5] to share information across pitchers $j = 1, 2, ..., J$.

$$\min_{\hat{w}_0, \hat{v}_j} \sum_{j=1}^{J} \left\{ \frac{1}{n^j} \sum_{i=1}^{n^j} (r_i^j - x_i^j.w_j^T)^2 + \frac{\lambda_1}{J}|v_j|^2 \right\} + \lambda_2|w_0|^2 \quad (6)$$

where $w_j = w_0 + v_j$.

Regularization parameters λ_1, λ_2 and shrinkage coefficients β are found by maximizing the area under the ROC curve for true class labels y_i^j (whether a run is given in the next inning) and the prediction scores r_j^j (estimated PTB) on the validation set.

4. EXPERIMENTAL RESULTS

We evaluate our method on the MLB data for years 2006-2010 from STATS Inc., which contains a record of each pitch thrown in both the regular and post seasons [6]. We train our model using the first 80% of the games of each season, and test it on the last 20%. We choose the regularization parameter and the threshold cut-off to binarize the regression outputs using cross validation on the training set.
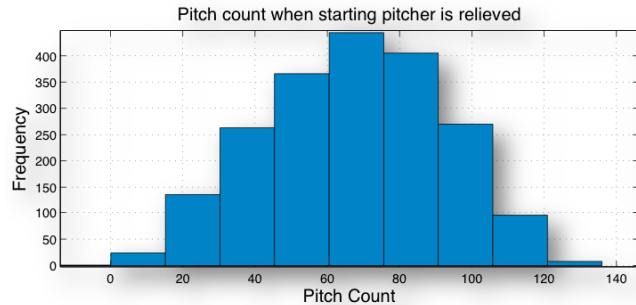

Figure 1: Histogram of the pitch counts when the starting pitcher is relieved

We first evaluate our model (Ours) relative to a model designed to mimic the way managers made decisions (Manager). We learn a manager model based on actual decisions of MLB managers. Managers often use pitch count (e.g., pitch count > 75) (Figure 1) and opposing team score to decide when to relieve the starting pitcher [2, 17, 12]. Our manager model learns pitcher-specific parameters that fit these two variables against the manager's decisions on the training data. For the 76 pitchers who pitched at least 500 pitches in each year, this heuristic model accurately models the manager's decision in 95%($\pm$1.2%) of the innings (F-score 0.87 ± 0.02), i.e., it is a reasonably accurate model of major league managers.

The manager model predicts whether or not a run will be scored in 75%($\pm$4.6%) of the innings. In contrast, our model makes a correct prediction for 81%($\pm$4.9%) of the innings. Figure 2 plots the histogram of accuracies for both methods. Notice that the methods are far more accurate for some pitchers than for others.

4.1 Useful features

The weights of a linear regression can be used to identify the most useful predictors. Table 2 lists the top 5 predictors based on the mean of the weights across pitchers ($|w_0|$).

Table 2: Top predictor weights across pitchers ($|w_0|$)

Pitcher-Inning prior (SLG)
Pitcher-Batting team prior (SLG)
Pitcher-1st batter (in the lineup) prior (Runs)
Pitcher-3rd batter (in the lineup) prior (Runs)
Pitcher-Inning prior (number of times)

Surprisingly, pitch count does not appear in Table 2. This does not mean that pitch count is irrelevant in predicting scoring. The inning (in the form of pitcher-inning prior based on SLG), a variable that correlates well with pitch count, turned out to be the most important predictor.
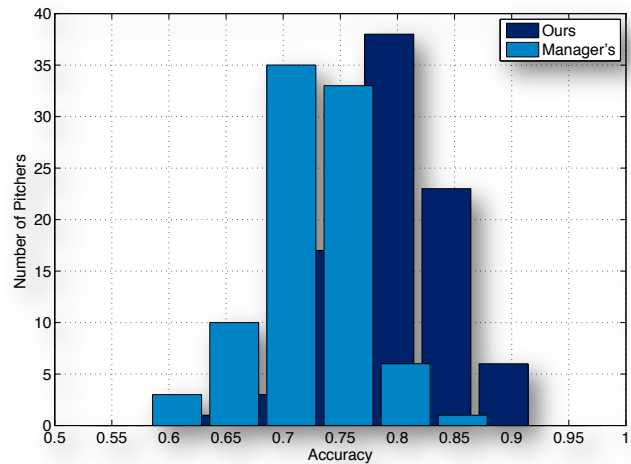

Figure 2: Model accuracy across pitchers

Table 3 lists predictors with the highest variance across pitchers based on the standard deviation of the weights across pitchers ($\sum_{j=1}^{J} |v_j|^2$). Notice that the last two features in this table also appear in Table 2. This emphasizes the importance of the role played by having both w_0 and v_j in Equation 6.

Table 3: Top predictor weights with high variance across pitchers ($\sum_{j=1}^{J} |v_j|^2$)

Pitcher-Inning prior (ERC)
Pitcher-home team prior (ERC)
Pitcher-batting team prior (ERC)
Pitcher-3rd batter (in the lineup) prior (Runs)
Pitcher-Inning prior (number of times)

Although features such as previous inning statistics (number of strike outs, ball and strike counts, etc.) do not appear in the top 5 list, they offer critical information in certain situations, as we will see in Section 5.

4.2 Predictability

Next, we test the predictabilities of starting pitchers in various situations. We use our method's accuracy to quantify the predictability. Figure 3 plots the average accuracy against the inning number across all the pitchers for our method and the manager model. The sizes of the circles denote the number of games.

At the beginning of the game, both methods have similar predictability. The difference increases drastically in later innings. As the game progresses, especially after the 4th inning (i.e., in the second half of a game), by exploiting the information in previous innings to track the progression of the game, our method becomes more accurate.

4.3 Evaluation

Since runs are scored in only about 10% of the innings pitched, accuracy is not the best measure of performance. A model that always predicts no-run innings would be accurate 90% of the time, but a prediction that always leads to leaving the pitcher in would be of no use. Therefore, we use the following set of measures for evaluation. The relationships
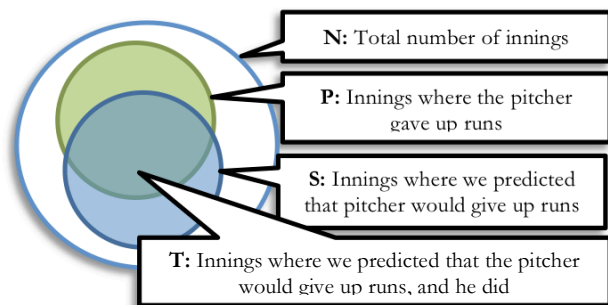
Pitcher	Accuracy(%)		F-Score		Odds Ratio	
	Ours	Manager's	Ours	Manager's	Ours	Manager's
Roy Halladay	80	79	0.48	0.19	1.3	0.88
CC Sabathia	89	80	0.50	0.13	5.5	0.86
Jon Lester	85	78	0.40	0.14	2.8	0.68
Average on 76 pitchers	81 ($\pm$4.9)	75 ($\pm$4.6)	0.41 ($\pm$0.1)	0.26 ($\pm$0.15)	3.2($\pm$1.2)	1.2 ($\pm$1.1)
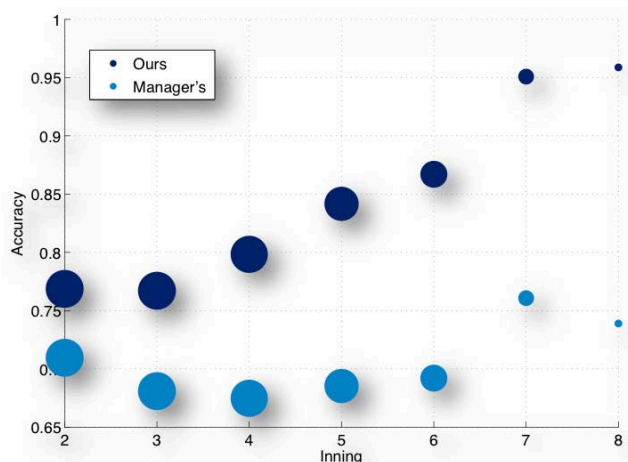
Figure 3: Accuracy of the models for each inning

among the variables used in these evaluation measures are depicted in Figure 4.

Figure 4: Evaluation measures

- Accuracy: mean number of correct predictions

- F-Score: harmonic mean of precision and recall

$$
\begin{aligned}
\text{F-Score} &= \frac{2 \times Precision \times Recall}{Precision + Recall} \\
Precision &= T/S \\
Recall &= T/P
\end{aligned}
$$

- Odds ratio: the odds of an inning with runs among the innings that are predicted to have runs versus the odds of an inning with runs among all the innings. An

odds ratio greater than one implies an improvement compared to random.

$$
\text{odds ratio} = \frac{T/(S - T)}{P/(N - P)}
$$

Table 4 presents the results for three well known pitchers and a summary across the pitchers. Our method outperforms the manager model in all categories. The difference in the average odds ratios is particularly striking. For our model, in those the innings where a pitcher is predicted to allow a run, the likelihood of him giving a run is 220% greater than his likelihood on all the innings. This improvement is only 20% for the manager model. The relatively poor performance of the manager model does not seem to be related to managers removing starting pitchers earlier to "save their arms," since on average our model removes pitchers sooner than does the manager model.

5. POTENTIAL IMPACT

In this section, we discuss the potential impact of using our method to decide when to remove a starting pitcher. We consider only the fifth inning on, since in the early parts of the game many factors other than expected effectiveness figure into the decision of whether to remove the starting pitcher. Out of 21,538 innings, our model disagreed with the manager's *actual decision* (i.e., not what our manager model would have done) a surprisingly high 66% of the time.

- There were 5,012 innings which the manager removed the starting pitcher and our model would reach the same decision.

- There were 6,201 innings in which the manager allowed the starting pitcher to continue and for which our model would reach the same decision. In roughly 17.7% of those innings the starting pitcher surrendered at least one run.

- There were 9,288 innings in which the manager elected to leave the starting pitcher in, but our model would make the opposite decision. In roughly 31.5% of those innings the starting pitcher surrendered at least one run.

- There were 1,037 innings in which the manager removed the starting pitcher and our model would have allowed the starter to continue. There is no way to know how the starter would have done had he not been removed.

Our model has a "quicker hook" than most MLB managers, i.e., it tends to remove starters earlier than is typical. Given the relatively poor performance of pitchers when they are left in despite what our model would suggest, there is

reason to believe that applying our model will lead to fewer runs being surrendered.

5.1 Case Study: Jul 19, 2010, Brewers at Pirates

Table 5: Runs scored in each inning

Inning	1	2	3	4	5	6	7	8	9
Pirates	0	0	0	0	1	0	0	0	0
Brewers	0	0	0	0	1	2	0	0	0

In the July 19, 2010 Brewers at Pirates game, the Pittsburgh Pirates changed their pitchers 4 times in the game (twice in the 7th inning). Table 5 lists the runs scored in each inning. Innings where a pitching change happened are highlighted. Jeff Karstens, the starting pitcher for the Pirates in that game, was allowed to pitch through the 6th inning and was relieved only after the damage was done. It is possible that if the pitcher change had happened earlier, the result of the game may have been different. We investigate whether our model would have recommended such a change.
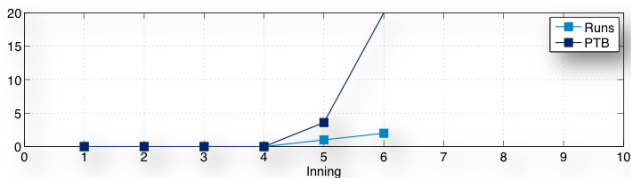

Figure 5: Runs given by Jeff Karstens in each inning

Figure 5 shows the runs and PTB given the starting pitcher. At first glance, it seems that we couldn't have predicted the sixth inning, because neither runs nor PTB of the previous innings (1-5) are indicative of what happened next.
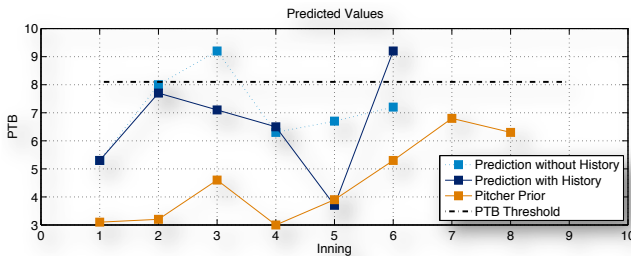

Figure 6: Runs predicted by our model for Jeff Karstens in each inning

Figure 6 shows the estimated PTB's produced by three different components of our method. Using only the pitcher-inning prior statistics (orange), i.e., pitcher's likelihood of giving up a run in an inning, we estimate the expected PTB to be 5.7 for the 6th inning. This is well below Jeff Karstens' pitcher specific cut-off $\hat{b}_j$ (PTB = 8.1). When other factors (e.g., pitcher-batter, pitcher-venue priors) are considered (light blue), estimated PTB rises to 7.6. However, this model wrongly predicts a run for the 3rd inning. This is because of a high prior for this pitcher for the 3rd inning.

Our final model (dark blue), which also incorporates previous innings' results, avoids the mistake in the 3rd inning, and correctly estimates a high PTB for the 6th inning.

Table 6: Game events

Inning	Bases Advanced	Events
1	0	Ground out, Flied Out, Flied Out
2	0	Flied Out, Ground out, Line Out
3	0	Flied Out, Strike Out, Strike Out
4	0	Pop out, Strike out, Strike Out
5	4	Home Run

Table 6 lists the factors that played an important role in our prediction. Since our model is also using the results from previous innings, our model gradually decreases the estimated PTB for the pitcher until the 5th inning, because he appears to be effective (zero bases advanced in the first 4 innings). The four strike outs in the previous two innings, which is indicative of the pitcher's skill, result in a drastic drop in the estimated PTB for the 5th inning. However, a home run in the 5th inning coupled with his high prior statistics for the 6th inning results in a reversion to a high estimated PTB for the 6th inning, which turns out to be correct in this case.

6. LIMITATIONS AND FUTURE WORK

There are several technical limitations in evaluating the real world application of our method. First, it is impossible to say what would have happened in those situations where a manager removed a pitcher that the model would have kept (i.e, we don't have gold standard), since we don't know whether or not the removed pitcher would have given up a run had he not been removed. Hence, our evaluations are one sided.

We consider only whether a run will be given in the next inning to decide whether to let the starting pitcher start the next inning. In practice, many other considerations come into play, particularly in early innings. Also, the starting pitcher is often removed in the middle of the inning. Our current method doesn't address this scenario. We intend to do pitch by pitch prediction in our future work.

We make no claim for the optimality of our choice of features. In fact, we expect that further study will lead to feature vectors that yield better performance.

We expect that our method can be applied to other similar sports. For instance, in cricket, we can use the same approach to predict the best bowler to bowl the next over.

7. CONCLUSION

Using information about the current at bat, game situation, and historical data, we estimate the number of runs given in the next inning. Using our method, MLB team managers can decide when a starting pitcher should be relieved. The results suggest that using our model might have led to better decisions than those made by major league managers.

For those games in which a manager left a pitcher in that our model would have removed, the pitcher ended up surrendering a run 60% of the time in the next inning, despite the fact that runs are scored in only about 10% of innings.

8. ACKNOWLEDGMENTS

References

[1] J. Albert. Pitching statistics, talent and luck, and the best strikeout seasons of all-time. *Journal of Quantitative Analysis in Sports*, 2(1):2, 2006.

[2] G. Baseball. The pitching rotation and the bullpen@ONLINE. `http://www.howbaseballworks.com/RotationandBullpen.htm`, Jan. 2013.

[3] B. Baumer and S. Ben. Why on-base percentage is a better indicator of future performance than batting average: An algebraic proof. *Journal of Quantitative Analysis in Sports*, 4(2):3, 2008.

[4] R. M. Bell and Y. Koren. Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 43–52. IEEE, 2007.

[5] T. Evgeniou and M. Pontil. Regularized multi–task learning. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 109–117. ACM, 2004.

[6] G. Ganeshapillai and J. Guttag. Predicting the next pitch. *Sloan Sports Analytics Conference*, 2012.

[7] A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[8] G. Huckabay. 6-4-3: Reasonable person standard@ONLINE. `http://www.baseballprospectus.com/article.php?articleid=1581`, Aug. 2002.

[9] Imaginesports. Glossary@ONLINE. `http://imaginesports.com/bball/reference/glossary/popup`, Jan. 2013.

[10] B. James. *Whatever happened to the Hall of Fame*. Free Press, 1995.

[11] J. Keri and B. Prospectus. *Baseball Between the Numbers: Why Everything You Know about the Game Is Wrong*. Basic Books, 2007.

[12] B. Prospectus. *Baseball Prospectus 2004*. Wiley, 2004.

[13] B. Prospectus. *Baseball Prospectus 2011*. Wiley, 2011.

[14] N. Silver. Introducing pecota. *Baseball Prospectus*, 2003:507–514, 2003.

[15] N. Silver. *The Signal and the Noise: Why So Many Predictions Fail-but Some Don't*. Penguin Group US, 2012.

[16] S. Sullivan. State of the art: The actuarial game of baseball@ONLINE. `http://www.contingencies.org/mayjun04/stat.pdf`, June 2004.

[17] F. Zimniuch and L. Smith. *Fireman: The Evolution of the Closer in Baseball*. Triumph Books, 2010.