

Leveraging Pitcher-Batter Matchups for Optimal Game Strategy

Paper Track: Baseball
Paper ID: 13603

Willie K. Harrison and John L. Salmon†*

*Department of Electrical & Computer Engineering, Brigham Young University

†Department of Mechanical Engineering, Brigham Young University

1. Introduction

Recent play in Major League Baseball (MLB) has showcased many attempts to achieve an advantage through smart selection of pitcher-batter matchups. One such case in the 2018 postseason had the Los Angeles Dodgers' manager, Dave Roberts, selecting an all right-handed batting lineup to face both Chris Sale and David Price in games one and two of the World Series. Both pitchers are left-handed, and the choice certainly tailors to the well-known lefty-righty handedness matchup strategy [10]. Furthermore, this choice was inline with the platoon system that the Dodgers had employed throughout the 2018 regular season [15], often starting all right-handed batting lineups against left-handed starters, seemingly with great success.

Although attempts have been made to optimize pitcher-batter matchup strategies in the sabermetrics [2, 7, 9] literature (e.g., [5, 6]), these approaches have tended to apply averages of large data sets in an attempt to fine tune some matchup data. For example, Hirotsu and Wright used the handedness (i.e., left handed or right handed) of both the pitcher and the batter to adjust average batting statistics in an attempt to optimize the choice of pitcher substitution [6]. In this approach, if a batter's dominant hand is opposite to that of a pitcher's dominant hand, then the offensive statistics of the batter are assumed to be enhanced slightly for that specific matchup. Also, if a batter's dominant hand is the same as that of a pitcher, then the offensive statistics of the batter are assumed to be degraded slightly for that specific matchup. The authors of [6] furthermore adjust these statistics based on team matchup data, assuming that above average offensive teams will have an enhanced advantage when facing below average defensive teams, etc. Although these adjustments are likely steps in the right direction towards optimal game strategy, the author's are unaware of any advanced individual pitcher-batter matchup data used to improve estimated results of the matchups.

Techniques such as those described in [6] attempt to compensate for the lack of statistically significant individual pitcher-batter matchup data in Major League Baseball by using average performance statistics in place of specific matchup statistics. To truly optimize with respect to matchup data, each pitcher-batter matchup should be tested in isolation until the data for that matchup is statically significant. Although handedness has proven to affect the overall statistics, there exist cases where this trend does not hold. That is, the statistics regarding some pitcher-batter matchups seem to defy the accepted truth that a batter is better off facing a pitcher with the opposite dominant hand as himself. A clearer

2019 Research Papers Competition
Presented by:

estimate of offensive and defensive output, therefore, requires an extensive amount of pitcher-batter matchup data, which unfortunately will never exist for all possible pitcher-batter matchups. This is always true in the case of rookie players, where the players simply have not been playing long enough to generate meaningful matchup data; but it also holds true for long-time players who may have been in opposite leagues (i.e., American or National Leagues) or divisions for much of their careers.

This paper outlines a novel technique for optimizing pitcher-batter matchups in a Major League Baseball game from the defensive (pitching) point of view. Although this knowledge is always useful, it becomes particularly important in must-win scenarios, e.g., towards the end of a regular season and during the postseason. We address two problems in this work: 1) how one can find an optimal pitching strategy for facing a set number of consecutive batters when statistically meaningful data is available for each pitcher-batter matchup, and 2) how one can repeat the process when certain potential pitcher-batter matchups have only occurred a small number of times (perhaps even zero times) in the real world.

The rest of the paper is outlined as follows. Section 2 motivates the research by investigating some early results of matchup specific optimizations over one inning of play. Section 3 presents the technical aspects of our techniques in detail. Full results of matchup optimizations with both dense and sparse data sets are given in Section 4. Finally, we offer conclusions of our work in Section 5.

2. Motivation

Before going to great lengths to devise techniques to optimize pitcher-batter matchups in a Major League Baseball game, we first need to motivate the idea by showing that optimizing a matchup strategy is worth a significant number of runs to a team. If all strategies are reasonably close to equivalent in terms of expected outcomes, or if following the lefty-righty handedness rule will deliver results that are essentially optimal, then there may be no need for additional research.

The intuition behind using individual matchup data to predict results in an MLB game follows from one of the themes discussed at the MIT Sloan Sports Analytics Conference in 2017 [12], where several professional athletes on panels expressed the desire for analysts to prescribe optimal strategies for the individual athlete, rather than the average athlete. For example, Jason Gore, an American professional golfer, told how a certain golf hole (a short par-4) should apparently be approached with a driver off the tee for the average player. Averaging over years of statistical data on that particular hole shows that players who tee off with a driver score better on the hole. However, Gore himself told how this approach would have been foolish for him to follow, since it would then likely require a second shot that is an obvious weakness in his own game [4]. Analyzing Gore's data alone draws this point out and prevents a poor strategic choice for him when playing the hole.

2.1. Importance of Matchups in Game Strategy

Figure 1 shows early results of this research that motivate additional investigation. Note herein that we have found a case with nine batters and five pitchers for which all 45 matchups have at least 30 at-bats between the 2000 and 2017 MLB seasons, inclusive [1]. The individual matchup data were then used to estimate probabilities of outcomes for each pitcher-batter matchup, and 500,000 innings

Table 1: Pitcher and batter names by index for the experiment portrayed in Figs. 1 and 2. Herein, R: right-handed, L: left-handed, B: switch hitter.

Index	Pitcher - Throws R/L	Batter - Bats R/L/B
1	Jeremy Guthrie - R	Elvis Andrus - R
2	Felix Hernandez - R	José Bautista - R
3	Jon Lester - L	Asdrúbal Cabrera - B
4	Rick Porcello - R	Melky Cabrera - B
5	Ervin Santana - R	Robinson Canó - L
6	-	Nelson Cruz - R
7	-	Edwin Encarnación - R
8	-	Paul Konerko - R
9	-	Joe Mauer - L

of play were simulated using a computer program for every possible pitching strategy against a fixed batting lineup. More on this technique is given in Section 3, but the important thing to note here is that the number of expected runs for this particular group of pitchers and batters ranges between 0.39 runs and 0.95 runs per inning. Thus, it appears on the outset that setting the matchups right can be worth more than half a run per inning! The strategies are labeled on the figure for both the best and worst cases, and indicate the pitcher chosen to face each batter for up to 10 consecutive batters.

For example, the optimal pitching strategy is shown in the figure to be {4333311111}. This means the 4th pitcher should face the lead off batter, and then be replaced by the 3rd pitcher to face the next four batters. If the inning is not yet complete, the 1st pitcher should then come in to face the next five batters. The pitchers and batters used for this simulation are given in Table 1, and clearly represent a hypothetical situation. Some of these players are currently retired, and this “game” was never actually played in real life. Pitcher and batter dominant hands are also included in the table, wherein we see that the optimal strategy follows the lefty-righty convention for handedness much of the time. The coloring of Figure 1 is chosen to show the pitcher that faces batter three (Asdrúbal Cabrera) in the simulation. It was determined by inspection after the simulation that this one matchup is the best single-matchup estimator for the overall scoring in the inning. Rick Porcello seems to be the worst pitching choice to face this batter, and Jon Lester is the best. Thus, defining a small number of matchups correctly may provide most of the benefit of the process.

2.2. Overcoming Sparse Matchup Data

The early results in Figure 1 are encouraging, but require a reasonably large number of matchups for all pitcher-batter combinations in the sets. Even finding cases like this in the historical data is challenging, and the sets of pitchers and batters tend to be comprised of both retired and active players. These shortcomings bring about an obvious question as to whether the technique can actually be used in practice. We investigated the correlation between all 45 matchups used to generate Figure 1, and performed a clustering algorithm based on these correlations. The results are shown in Figure 2, where we see that one can form meaningful clustering so as to group matchups into “types.” Since this is the case, it appears that not all matchups are as different as they may seem, and classification of a matchup

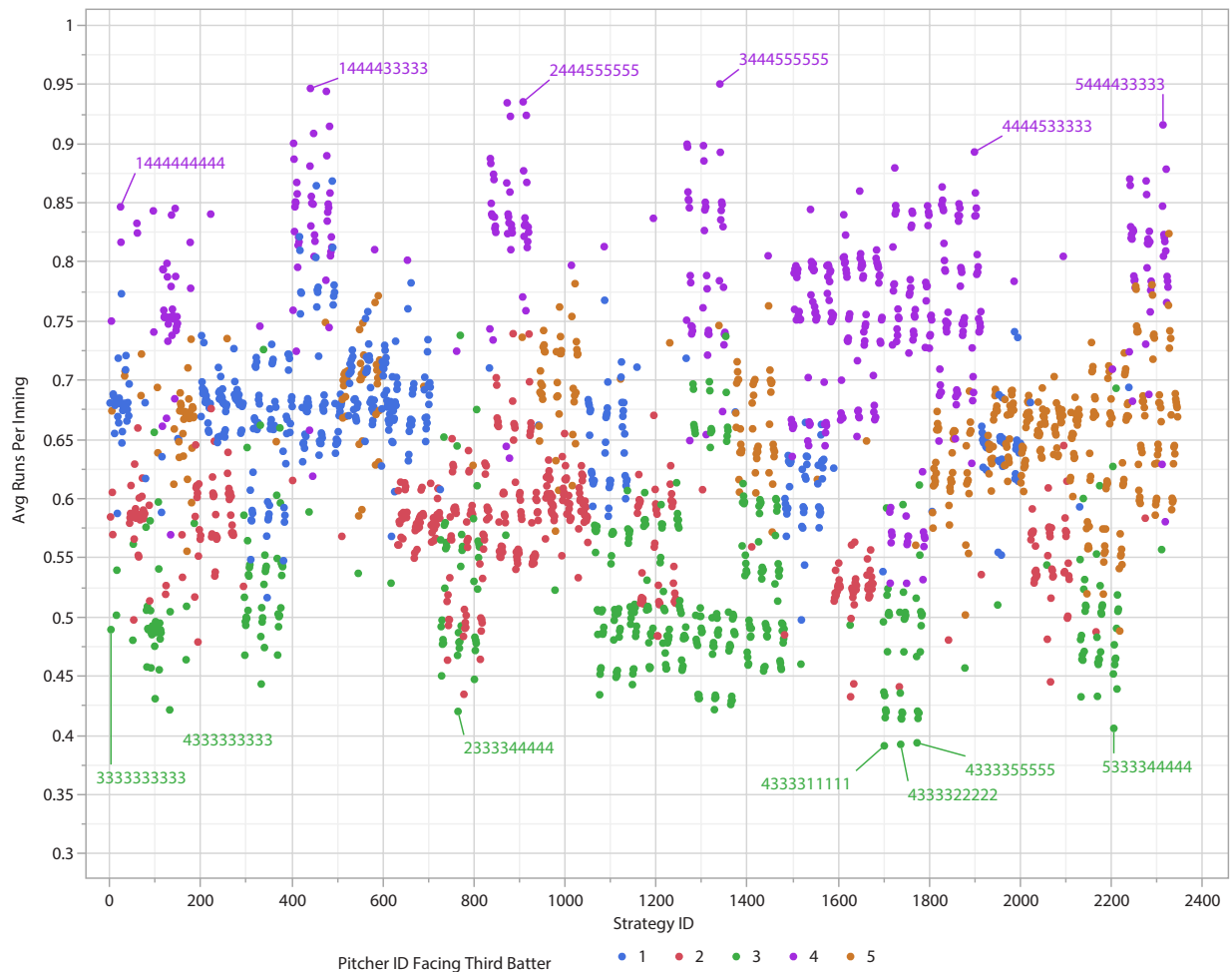


Figure 1: Average number of runs per pitching inning for each strategy with optimal and worst strategies labeled. Labels are for Pitcher ID assignments for 10 batters.

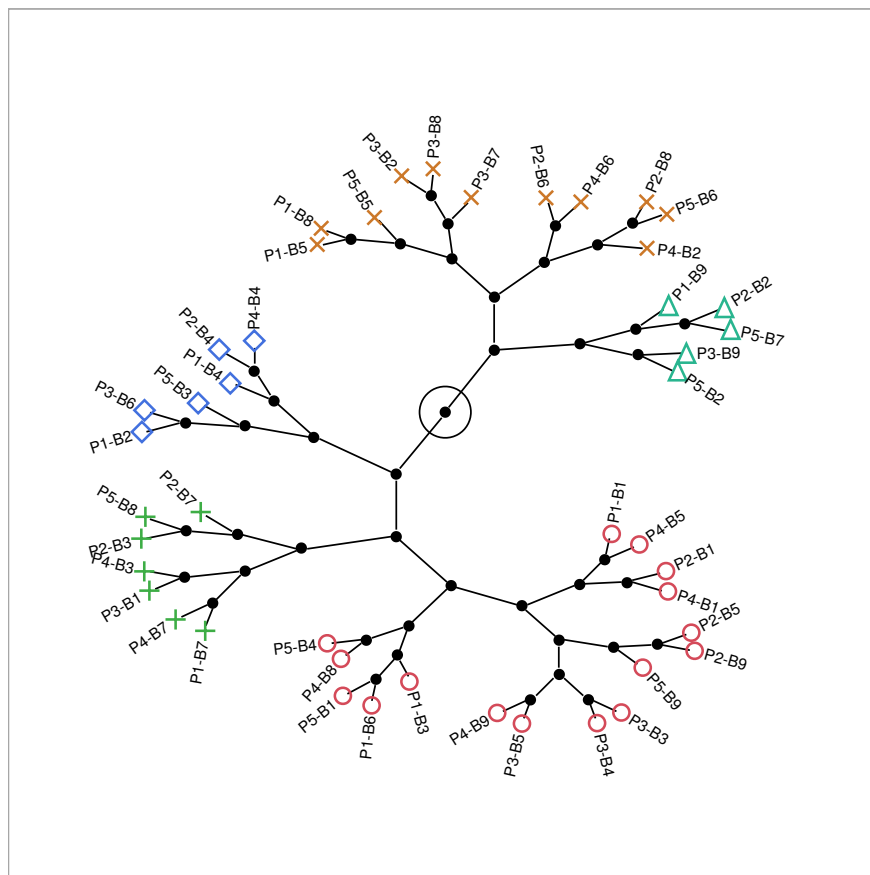


Figure 2: Constellation plot demonstrating clusters among the 45 different pitcher-batter (labeled as P_x-B_x) matchups.

into a proper type can then allow the type statistics to take the place of the matchup statistics in game simulation. Doing so allows one to estimate an optimal pitching strategy based on matchups, even when matchup data is sparse. The open questions are: 1) how many types should be chosen, 2) how should the decision algorithm be structured to properly classify sparse-data matchups into one of the established types (so as to minimize error in the estimate), and 3) how close is the estimate to the truth. These questions are addressed in Section 3.3.

3. Methods

Although the initial results in Section 2 used only regular season matchup data from 2000-2017, for the rest of this paper we use a slightly larger data set that includes 2000-2017 postseason data as well as 2000-2018 regular season data. The event-level data is from retrosheets.org [1], and consists of more than 3.59 million matchup results. We screen these data to include only active players in the 2018 MLB season. The main reason we select our data in this way is to enable simulation of the actual matchup scenarios that occurred in the 2018 postseason as a special case in Section 4. Thus, we include only

data pertinent to active players that would have been available to MLB teams in October of 2018. We define statistically significant matchups as those occurring at least N times in our data set, and we'd like to find data structures that maximize N .

3.1. Finding Bicliques in Big (Matchup) Data

In order to validate our clustering techniques, we need to locate data sets that are extensive enough to act as the “truth” for our algorithms. Although we have no way of knowing the exact truth without thousands of data points for each matchup, we would like to find pitcher and batter sets where all matchups have occurred as many times as possible. We can then use the maximum-likelihood estimates for probabilities of events for each matchup, which amounts to taking the ratio of occurrences of the event (say a single, or a strike out) divided by the total number of matchups. Matchups that have occurred at least $N = 30$ times are reasonably significant from a statistical point of view [13].

Let G signify a simple bipartite graph with bipartitions on the vertex set of V and U . Here V is a set of vertices that is comprised of pitchers who have faced at least one batter N or more times, and U is a set of vertices or nodes that is comprised of batters who have faced at least one pitcher N or more times. Edges in G connect pitchers and batters who have faced each other at least N times.

In order to simulate a game of baseball using real matchup data that is statistically significant, we need at least nine batters in U such that all nine batters have faced the same set of pitchers at least N times. In other words, we need to find a subgraph of G , say G' , with at least nine vertices in $V' \subseteq V$ that forms a complete bipartite graph with a set of pitchers $U' \subseteq U$. This structure is called a biclique or a complete bipartite subgraph of G in graph theory [3], and finding bicliques of maximal size with efficient algorithms has been an area of research since the 1960s [11, 16]. We employ an adaptation of the technique in [16], that was developed with applications in genome research in mind, to find a maximum biclique in our pitcher-batter matchup data in the following sense. We desire at least $|U'| = 5$ pitchers to choose from to form optimal strategies, with the number of batters in the biclique $|V'| \geq 9$. Note that $|\cdot|$ indicates the cardinality of a set. We search for the biclique with $|V'| \geq 9$ and $|U'| = 5$ that maximizes N , the number of matchups required for a pitcher-batter pair to be connected with an edge in G [3].

The algorithm is straightforward given $G(U, V)$. We cycle through all possible choices of U' such that $|U'| = 5$, and calculate the intersection of the neighborhoods of all vertices in U' , where the neighborhood of a vertex is the collection of vertices connected to that vertex by an edge in G . If this intersection consists of at least nine vertices in V , then we call it V' , store U' and V' as a biclique matching our requirements, and continue to search for additional bicliques. We run this algorithm for decreasing N , until the program returns at least one biclique. For our dataset, we found exactly one biclique with five pitchers and 10 batters when $N = 36$. No bicliques exist with at least five pitchers and nine batters with $N > 36$. The biclique found is given in Table 2. Notice similarities in the biclique in Table 1 with this new biclique. This should not be surprising, as we are looking for sets of players that have played long enough to face sets of opponents at least N times. Players tend to be All-Star veterans in bicliques such as these.

Table 2: Biclique of active players in 2018 that maximizes the number of occurrences for each matchup at $N = 36$ with five pitchers and 10 batters. Herein, R: right-handed, L: left-handed, B: switch hitter.

Index	Pitcher - Throws R/L	Batter - Bats R/L/B
1	Felix Hernandez - R	Robinson Canó - L
2	David Price - L	Brett Gardner - L
3	Ervin Santana - R	Curtis Granderson - L
4	James Shields - R	Ian Kinsler - R
5	Justin Verlander - R	Nick Markakis - L
6	-	Joe Mauer - L
7	-	Adrián Beltré - R
8	-	Elvis Andrus - R
9	-	Nelson Cruz - R
10	-	Dustin Pedroia - R

3.2. Pitcher-Batter Matchup Strategy Simulation

There are many ways of generating estimates of probabilities of events for specific pitcher-batter matchups. One technique was just presented in the previous section, and requires one to find all matchup data for the pitcher-batter matchup in question, and form probability estimates using the maximum-likelihood ratio-of-occurrences rule [13]. If the pitchers and batters in question have faced each other a large number of times, then these estimates tend to be fairly reliable, and can be used to simulate innings of baseball, and hence, to classify pitching strategies in terms of matchups. Another technique for estimating event probabilities in a matchup is laid out in the section following, where probability estimates are made without a significant amount of matchup data. In this section, we discuss our technique for simulating innings so as to rank pitching strategies in terms of their ability to prevent runs by the opposing batting lineup. We use the biclique from Table 2, and omit the 10th batter to create a fictional nine-batter lineup.

Let the set of available pitchers in a bullpen be $P = \{p_1, p_2, \dots, p_{|P|}\}$. A pitching strategy is a sequence of numbers that indicate the index of the pitcher to face each batter. That is, let

$$\mathcal{S} = \{s_1, s_2, \dots, s_{|\mathcal{S}|}\} \quad (1)$$

signify a pitching strategy for $|\mathcal{S}|$ batters in a single inning. Each s_i for $i = 1, 2, \dots, |\mathcal{S}|$, must be in $\{1, 2, \dots, |P|\}$, and identical integers in $\mathcal{S}$ must occur consecutively (since replacing a pitcher makes him ineligible to pitch for the rest of the game by the rules of baseball [14]). By way of example, the strategy $\{33111\}$ indicates that pitcher p_3 should pitch against the first two batters of the inning, and then pitcher p_1 should face the next three batters. Note that the strategy is to be formed before play begins for the inning, although real managers could decide to update strategies many times during an inning of play.

The scenario we simulate is a single inning of play with up to $|\mathcal{S}| = 10$ consecutive batters. A simulated inning is deemed complete if three outs are made, or if $|\mathcal{S}|$ batters have been faced. In our simulations, the percentage of innings where 10 batters were faced without recording three outs is roughly 0.39%, meaning less often than 1 out of 250 innings simulated. We arbitrarily restrict the

Table 3: Possible outcomes for each at-bat. Probabilities for each of these events must be estimated for each pitcher-batter matchup to simulate innings.

HR: home run	T: triple	D: double
S: single	W: walk	FO: fly out
GO: ground out	K: strike out	HBP: hit by pitch

“manager” to using no more than three pitchers per inning, but this could be likewise adjusted. Consider, for example, the case where a manager is trying to decide whether to replace a pitcher or not. The manager could consider strategies of up to two pitchers, where one of the pitchers is the current pitcher.

Our simulations step through all possible pitching strategies that use three or less pitchers to cover up to 10 batters in an inning, and we simulate 500,000 innings for every possible pitcher-batter strategy under these constraints. There exist 2345 different strategies, for these constraints, and each can be compared against all others using the simulation technique.

Within play, probability estimates of matchup events are used to simulate outcomes of at-bats by generating random numbers and fitting them to the probability mass function of possible outcomes. At-bats can result in the set of outcomes given in Table 3. We let $P(X)$ signify the probability of event X occurring, where X takes on all possible outcomes in the table.

Rules of the simulation include the following:

- the batting lineup is fixed,
- up to three pitchers may be used to form a pitching strategy,
- runners on base advance on a flyout with probability 0.0715*,
- runners on base take an extra base on singles and doubles with probability 0.3356*,
- double plays are turned successfully with probability 0.6892*,
- the lead runner is put out with probability 0.5 in an unsuccessful double play attempt, else the batter is put out,
- base runners advance on ground balls with probability 0.25,
- no bases are stolen,
- no errors, wild pitches, or passed balls occur, and
- base runners are not picked off or thrown out going for extra bases.

The symbol * is used to indicate event probabilities that are estimated from the event-level historical data [1]. All other rules of simulation are set for ease of simulation, and many simplify the game to the point where a computer can “play” an inning of baseball using a random number generator. Results of this inning simulation technique are highlighted in Section 4 for the biclique in Table 2, and other scenarios encountered in the MLB postseason during 2018. Figure 1 shows the results of our simulation techniques using the biclique from Table 1.

3.3. Clustering for Classifying Pitcher-Batter Matchups

Since the aforementioned model and simulation require statistics for the individual pitcher-batter matchups, an algorithm is necessary for classifying those matchups for which the data is sparse (i.e. a small number of at-bats for a given pitcher-batter pair).

This is performed by analyzing the pitcher-batter matchups for which there are a large number of at-bats for the pair. We identified 5170 pitcher-batter matchups for which there were 35 or more at-bats for the pair during the last 19 years. A clustering algorithm, applying Ward's Method [8], is implemented which calculates the "distance" between each of these 5170 pitcher-batter matchups and sorts them in a sequence such that any pitcher-batter matchup is positioned beside its next-closest matchup in terms of correlation. The output of such a process is the dendrogram presented in Figure 3. We lump doubles and triples together and ignore HBP when performing the clustering to prevent the algorithm from building cluster types for rare, and somewhat random, events (i.e., T and HBP).

Figure 3 indicates 15 groupings which generally discriminate the full matchup space into clusters. Therefore, each individual matchup is closer to matchups within its own cluster than any other matchup outside of its cluster. The statistics from the 5170 pitcher-batter matchups are likewise presented in the form of a cell plot beside the dendrogram of Figure 3. The seven statistics used in the clustering algorithm are shown with higher probabilities in red and lower probabilities in blue. Groupings can be seen in this cell plot which are expressed into the clusters calculated on the right dendrogram. For example, the first cluster at the top (shown in red), has a relatively high probability of singles, $P(S)$, for the matchup. The last cluster, at the bottom, has a high probability of walks, $P(W)$. The other matchups between these will have different relative probabilities for homeruns, strike-outs, ground-outs, fly-outs, and multiple-base hits (doubles or triples). These data can be seen in various groupings or types within the cell plot with their corresponding section of the dendrogram on the right.

This can also be seen in the constellation map of Figure 4 where each matchup location is represented in a tree-like structure with the relative distance between matchups found through traveling to a common link or node. For example, a matchup found in the green cluster (bottom right) is closer to matchups in the red cluster (also bottom right) through a node which is only one step away from the root node (indicated with a large circle). Thus these red and green matchups are more similar to each other than they are to matchups in the blue cluster (top left) which is many nodes away.

The statistics for each cluster are presented in Figure 5 through a parallel plot which presents the "statistical thumbprint" of a particular cluster. Since each cluster is composed of 100s of matchups, the plots show the full range of statistics assigned to that cluster. However, careful observation of these plots reveal cluster differences that can be used to classify matchups outside of the initial training set (i.e., the 5170 pitcher-batter matchups). For example, the matchups of Cluster 12 are characterized by a high probability of strike-outs and low probability of doubles or triples, $P(DT)$, but with a somewhat higher probability of home runs, $P(HR)$. Cluster 13 is similar to Cluster 12 but with a lower probability of home runs. In contrast, Cluster 4 has a higher than normal probability of home runs and multiple-base hits but with low $P(K)$ and thus favors the hitter in the relationship. Other clusters have high probabilities of fly-outs and ground-outs, such as Clusters 8 and 2, respectively, while one cluster has a high walk probability, $P(W)$. Other clusters retain different combinations of high probabilities across the seven statistics including singles $P(S)$, fly-outs $P(FO)$, ground-outs $P(GO)$, and strike-outs $P(K)$.

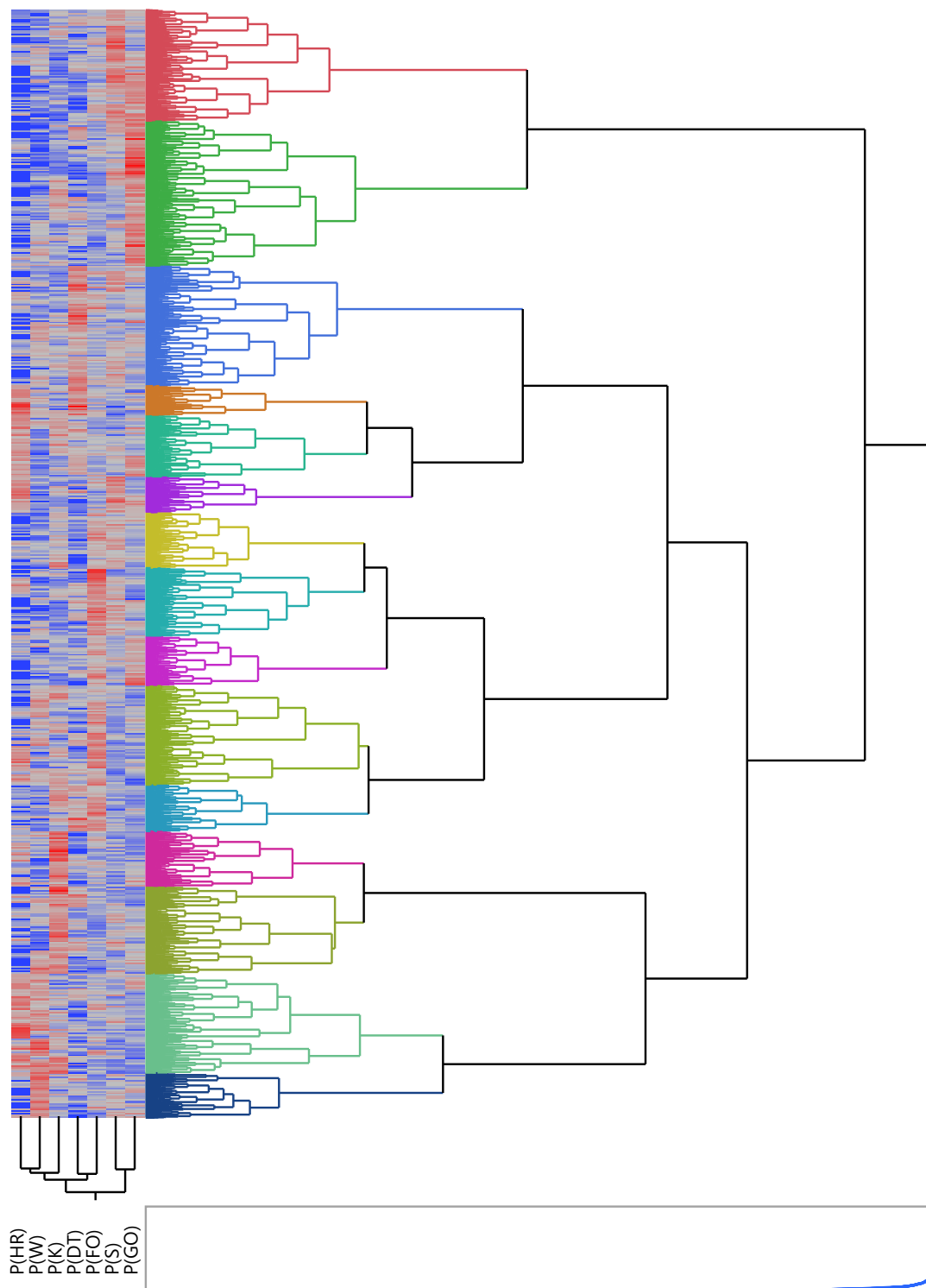


Figure 3: The resultant dendrogram from the clustering algorithm of 5170 high at-bat matchups. Different colors discriminate between the 15 clusters identified. Each matchup is closer to matchups within its own cluster than any other matchup outside of its cluster.

2019 Research Papers Competition

Presented by:

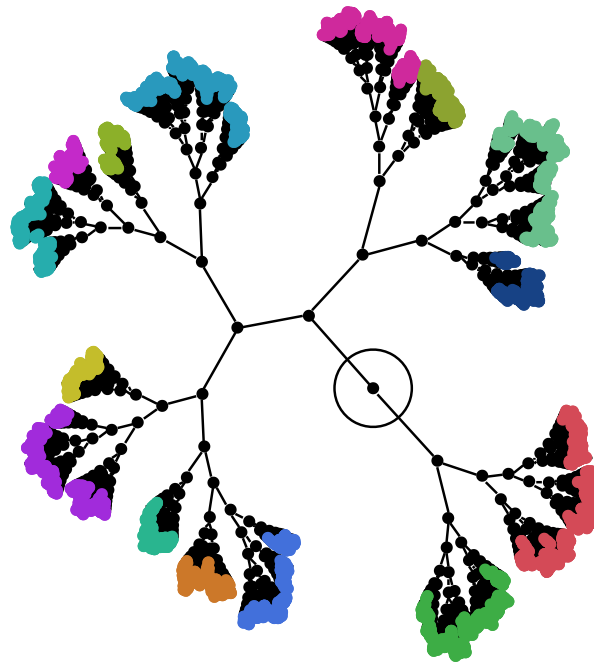


Figure 4: Constellation map presenting the tree-like structure for classifying the matchups into 15 clusters through relative distances of the statistics.

Ultimately, each matchup can be classified into one of the 15 matchup types defined by these clusters. Ideally, the statistics of a matchup can be extracted from the actual data, but since the number of at-bats for many matchups are insufficient, the “matchup type” may be useful to predict the result for a more accurate simulation.

The specific number of 15 cluster types was selected based on a comparison of the parallel plot of the 16th cluster with those of the other 15. Since no discernable difference to some of the other plots was observed in defining these matchups, the law of diminishing returns appears to have been reached. The selection of 15 clusters was considered to be the best balance between a sufficiently large set of clear matchup types while keeping the number of matchups per cluster high to maximize significance.

Thus, the remaining 823,594 matchups with low numbers of at-bats, which are not used in the creation of the clusters, can be classified according to their closeness to these 15 types. This process is summarized in Figure 6 where a subset of training data is presented in the left-most plot using two notional statistics. In this case, the clustering algorithm finds three clusters and designates them as 1, 2 and 3 respectively with red circles, green triangles, and blue diamonds as shown in the middle plot. The mean value for the particular cluster across the two statistics is represented with a filled in circle, triangle, or diamond on the far right plot with the cluster number indicated. The three matchups, A, B and C, which were not used in the clustering process are assigned a cluster based upon their “distance” to the mean value in each cluster. Therefore, A is assigned to Cluster 3, B is assigned to Cluster 2 (even

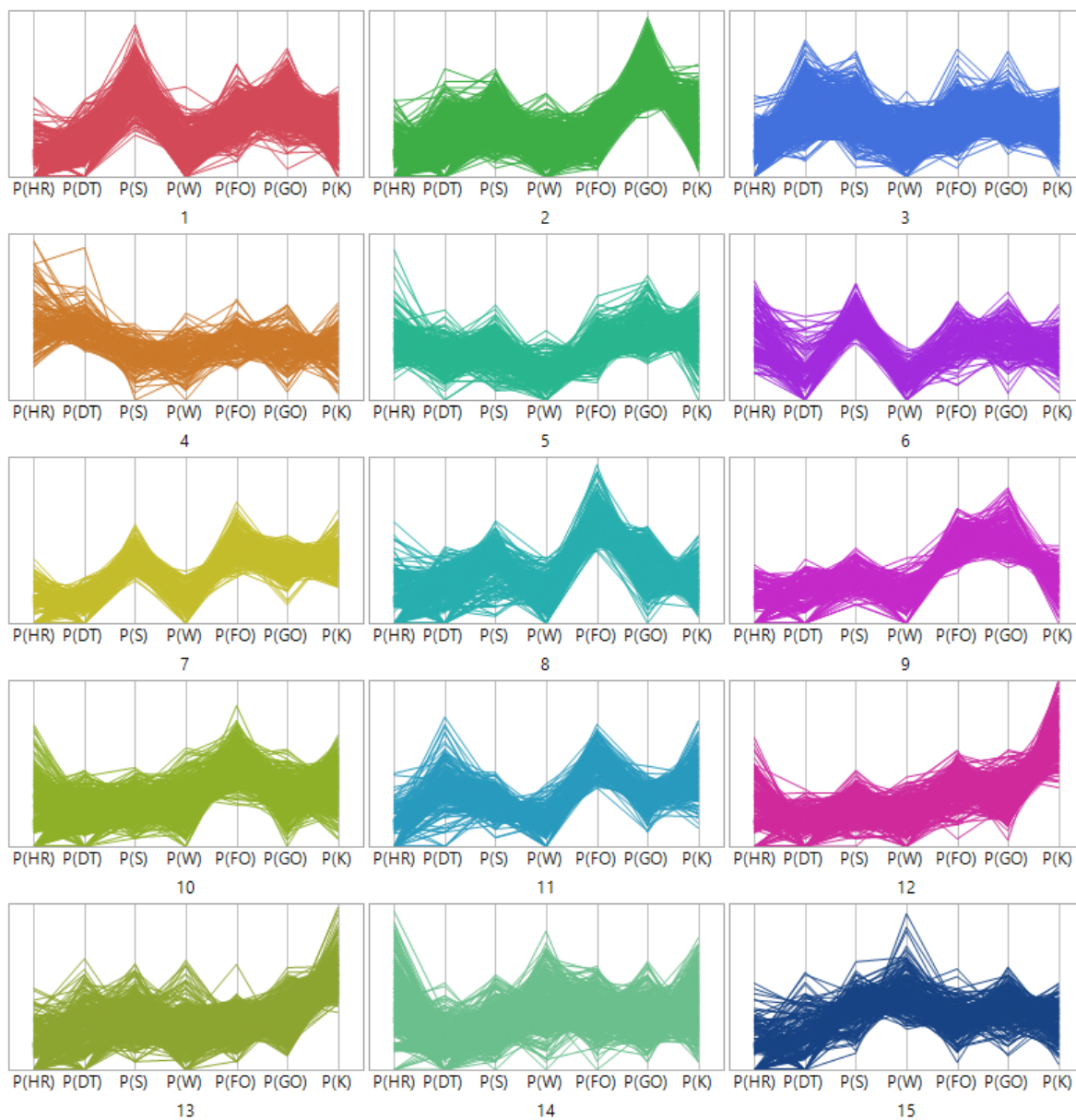


Figure 5: The probability statistics of each of the 15 clusters presented in parallel plots

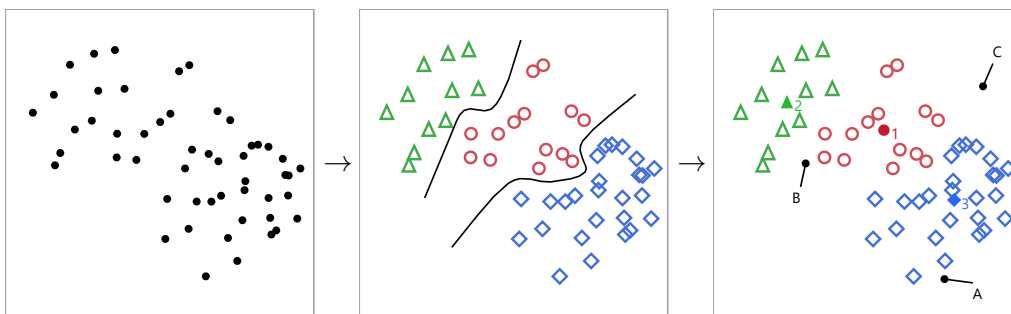


Figure 6: Generalized schema for the clustering algorithm and classifying of pitcher-batter matchup types using notional data for two statistics

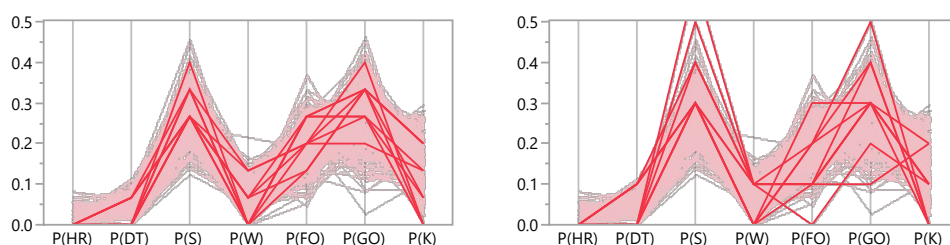


Figure 7: Validation examples of matchup type classification algorithm with 15 and 10 at-bats for the matchups shown, respectively, on the left and right. Shaded bands in the background show cluster tendencies from the training dataset.

though it could be closer to some individual matchups of Cluster 1), and C is assigned to Cluster 1 since it is slightly closer to the Cluster 1 mean than the Cluster 3 mean.

This classification is applied to all the matchups, even those with only one at-bat, to define all 828,764 matchups from the last 19 years. This classification process can be likewise explored by comparing the new matchups statistics to the cluster training set. The left plot in Figure 7 shows the statistical probabilities of 10 matchups, all with only 15 at-bats, overlaid onto the training data for Cluster 1. These data align well with the clustering statistics, validating the performance of the clustering and classification algorithms described above. The plot on the right in Figure 7 shows a similar result but with only 10 at-bats per matchup occurring during the last 18 years. The alignment and correlation is positive but the alignment is beginning to break down due to a low sample size. However, compared to all the other clusters this is still the best classification with the highest correlation and therefore each of these 10 matchups remain classified as Type 1. In other words, the distance from these matchups to the cluster means for all the other clusters is expectedly much larger. Therefore, this is the best classification that can occur based on the low number of at-bats for these specific matchups.

4. Results

In this section, we test our clustering techniques against “truth” data, and calculate correlations. We also compare our results to the case where only handedness determines the matchups. Finally, we end

with two case studies, and offer analysis of two playoff teams from 2018 when pitching in a must-win scenario.

4.1. Verification of Clustering

Although we label the maximum-likelihood estimates based on matchup data as “truth,” these estimates could clearly be improved with additional occurrences of each matchup. In essence, our clustering technique allows us to do this, as long as matchups can be sorted accurately into their types. However, we should be careful to verify that strategies are consistently classified, e.g., as good or bad, regardless of the approach taken.

In Figure 8, we show simulation results of 500,000 innings for all pitching strategies using three different sets of probabilities, allowing us to compare and contrast the three techniques. The “truth” data in the figure uses the actual matchup statistics to estimate event probabilities using the maximum-likelihood rule, and provides the first set of probabilities. The second set are derived from the 15 cluster type data, where each matchup is sorted into its closest cluster type, and then the cluster type statistics provide estimates for the probabilities. We note that the two approaches appear to match each other well across all pitching strategies in Figure 8. The consistency of the ranking of strategies is further verified in Figure 9, where the average runs per inning in the 15-clusters data is plotted against the same quantity when the truth data are used to form probabilities. A correlation coefficient of 0.9286 is calculated between these two data sets, thus verifying that the clustering technique preserves the matchup data sufficiently to identify good, bad, and mediocre pitching-batter matchup strategies.

Following lefty-righty matchup tradition provides the third approach to estimating the needed probabilities for simulating innings of baseball. We sort all matchups into only two clusters for this approach: 1) when the dominant hands of pitchers and batters are identical, and 2) when the dominant hands of pitchers and batters are opposite. When a batter is a switch hitter, we assume he bats with the opposite handedness of the pitcher. We average all matchups according to only these two clusters, and then use these probabilities to simulate innings of baseball. These are shown in Figure 8 as lefty-righty cluster type data. Here we see that sorting only according to handedness destroys much of the validity of the simulation. Figure 9 also includes the lefty-righty cluster type data plotted against the truth data, and we see that the correlation between the two simulations is much smaller than in the 15 cluster type data case. The correlation coefficient of these two data vectors is calculated only to be 0.4666. Certainly this indicates that the handedness of opponents has something to say about matchup results, but we can capture almost *all* of the subtleties of matchup strategy if we allow for 15 cluster types.

4.2. 2018 Postseason Test Cases

Although we assume that some Major League teams are already taking advantage of matchup data where it exists, we are, however, uncertain of how teams are currently dealing with cases where pitcher-batter matchup data are sparse or nonexistent. As a first step towards analysis of Major League teams, and their ability to adequately choose good pitching strategies, we take two cases from the 2018 postseason. Since our approach does not yet take into account multi-game scheduling, we look at must-win scenarios only, judging that winning an elimination game is a case where all available resources should

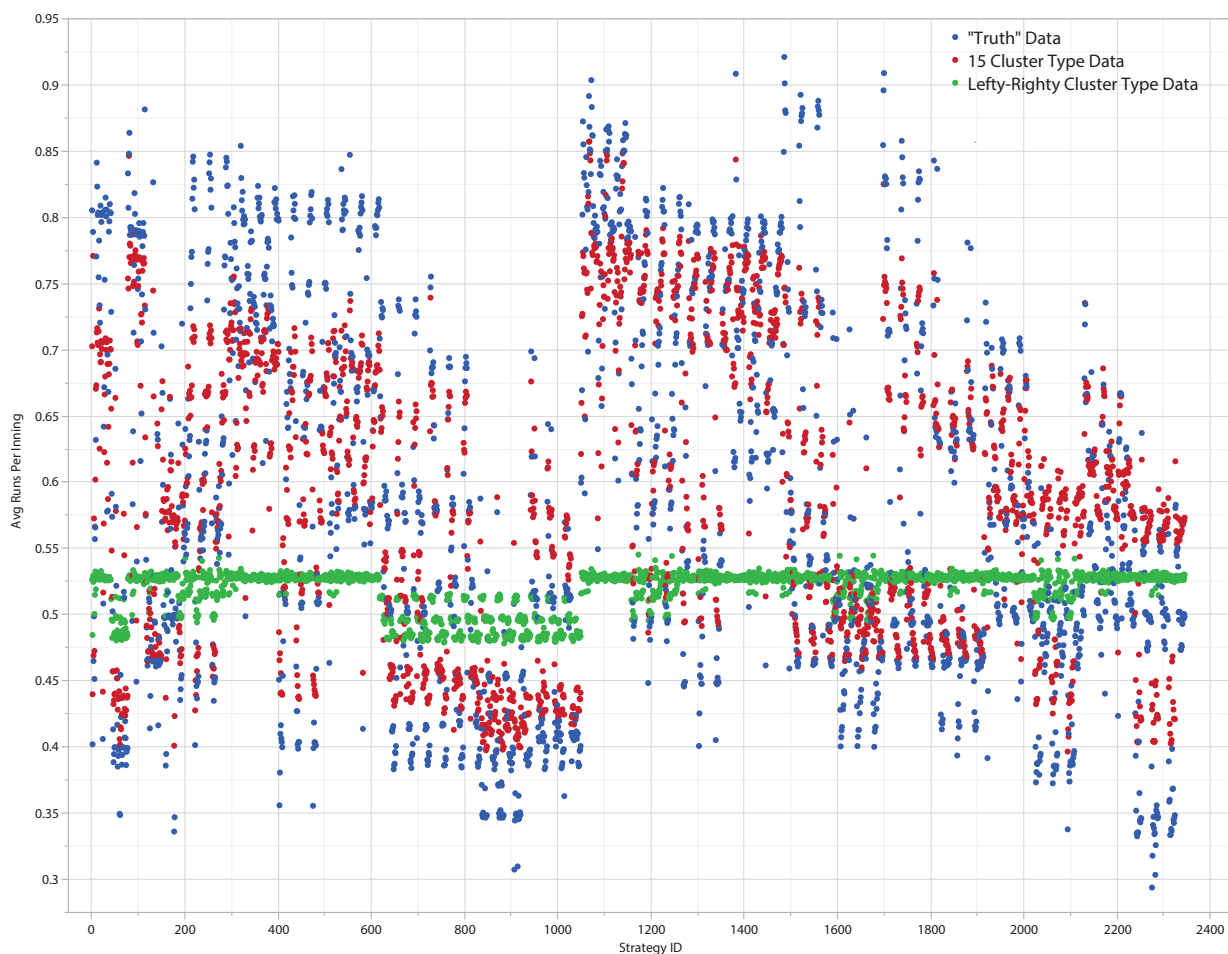


Figure 8: Average number of runs per inning for each pitching strategy. The “Truth” data (blue circles) indicate the expected runs from the simulations using the original biclique probability statistics while the “15 Cluster Types” data (red circles) use cluster means, and “Lefty-Righty Cluster Type” data (green circles) employ the lefty-righty clustering.

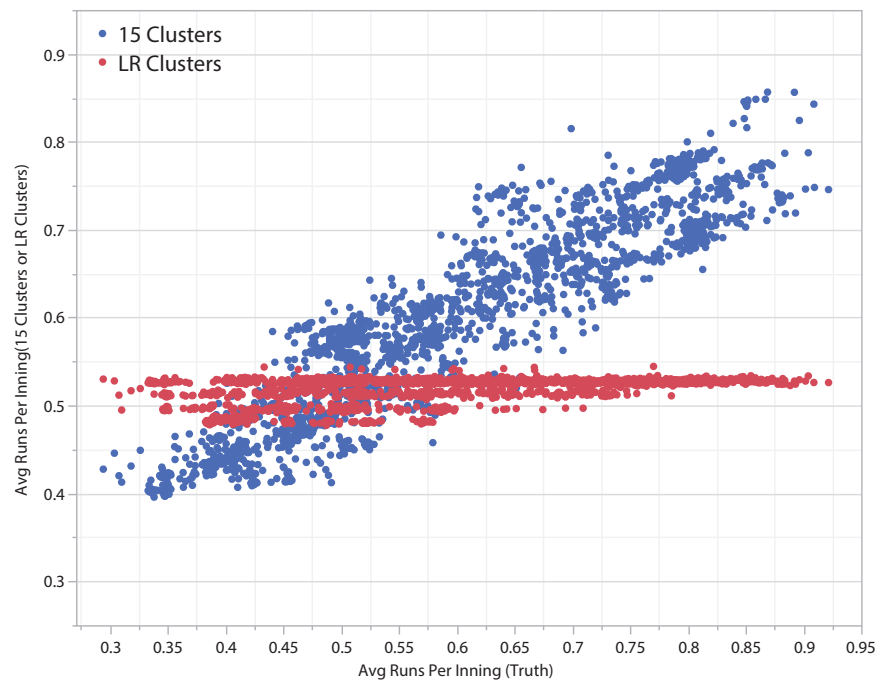


Figure 9: Comparison of the three approaches to estimating event probabilities used in the simulations from Figure 8. Using the “15 Clusters” data provides significantly higher correlation (0.93) between the expected runs per strategy from the “Truth” data as compared to the “Lefty-Righty Cluster Type” data (0.47).

Table 4: Selection of five New York Yankees pitchers and the opposing Boston Red Sox lineup for the 9th inning of Game 4 in the ALDS 2018. Herein, R: right-handed, L: left-handed, B: switch hitter.

Index	Pitcher - Throws R/L	Batter - Bats R/L/B
1	Aroldis Chapman - L	Steve Pearce - R
2	Chad Green - R	J. D. Martinez - R
3	Jonathan Holder - R	Xander Bogaerts - R
4	Lance Lynn - R	Ian Kinsler - R
5	J. A. Happ - L	Eduardo Núñez - R
6	-	Jackie Bradley Jr. - L
7	-	Christian Vázquez - R
8	-	Mookie Betts - R
9	-	Andrew Benintendi - L

be utilized to prolong a team's season. As such, we even consider starting pitchers as being available for relief pitching in some cases.

For the sake of even comparisons, we investigate two elimination games against the Boston Red Sox. One with the New York Yankees in the American League Division Series (ALDS), and one with the Houston Astros in the American League Championship Series (ALCS). For each case, we investigate the exact scenario in the top of the 9th inning when both the Yankees and the Astros trailed in their respective elimination games and compare their pitching strategies to the optimal ones. We should be careful to point out that pitching strategies in real games are chosen in real time, and real-time decisions are always conditioned on the state of the game. We produce strategies for the entire inning, and do not rerun our simulations at any point in the inning conditioned on actual events occurring. Thus, it is possible that decisions later in the inning may be better or worse than we indicate in our results. Optimal real-time decisions would require us to rerun our simulations after each at-bat.

4.2.1. Boston Red Sox vs. New York Yankees - Game 4 ALDS

The Boston Red Sox led the New York Yankees 4-1 in the top of the 9th inning in Game 4 of the ALDS in 2018. If the Yankees lost this game, their season would be over. Let us see how well the Yankees optimized their final inning of pitching in 2018. The opposing Red Sox lineup, starting with the lead-off batter in the 9th inning, and a selection of five Yankee pitchers who were available to pitch that inning are given in Table 4. Figure 10 shows the average number of runs that the Red Sox were expected to score in the 9th inning as a function of pitching strategy, highlighting some of the best and worst cases.

Although we present 10-batter strategies, as we did earlier in the paper, the Yankees selected Aroldis Chapman, Pitcher 1, to pitch the 9th inning, and he made three consecutive outs. Thus, we can only verify the correctness of the pitching strategy up to the first three batters. For the first three batters only, we see in Figure 10 a nice strategy consistent with this choice of {111255555}, and so we must give the Yankees high marks, even if the choice of pitching Chapman seemed like an obviously good

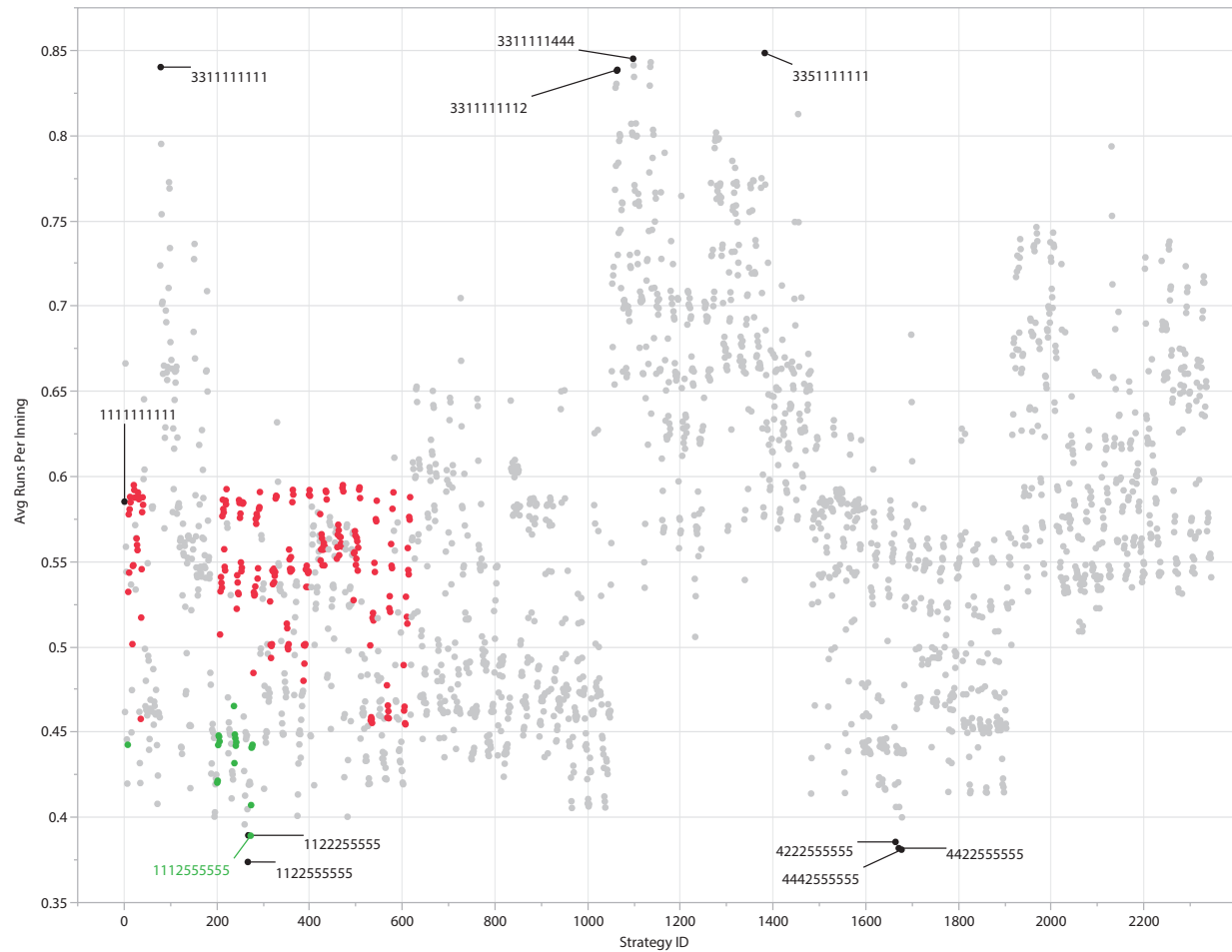


Figure 10: Average number of runs per inning for each pitching strategy with optimal and worst strategies labeled for the final inning of the 2018 American League Division Series. Labels are for Pitcher ID assignments for 10 batters.

Table 5: Selection of five Houston Astros pitchers and the opposing Boston Red Sox lineup for the 9th inning of Game 5 in the ALCS 2018. Herein, R: right-handed, L: left-handed, B: switch hitter.

Index	Pitcher - Throws R/L	Batter - Bats R/L/B
1	Gerrit Cole - R	Jackie Bradley Jr. - L
2	Dallas Keuchel - L	Mookie Betts - R
3	Roberto Osuna - R	Andrew Benintendi - L
4	Ryan Pressly - R	J. D. Martinez - R
5	Héctor Rondón - R	Xander Bogaerts - R
6	-	Mitch Moreland - L
7	-	Ian Kinsler - R
8	-	Rafael Devers - L
9	-	Sandy León - B

choice. We see that Chapman was, in fact, the optimal choice to start the inning, but only for the first two batters, although letting him pitch to the third batter didn't give much advantage to the opponent. We also see a number of other reasonably good strategies that do not include Chapman at all, e.g., {444255555}.

Let us ask the question, however, would the "all-Chapman" approach have been the best, given the other four pitchers that were available, if more than three batters had been faced in that inning? The all-Chapman strategy {111111111} is clearly a suboptimal choice, as labeled in the figure at the far left. Furthermore, the green data in the figure show all strategies consistent with the first four batters being faced by {1112}, while the red data in the figure show all strategies consistent with the first four batters being faced by {1111}. After three batters, sticking with Chapman would have presented more of a risk than having him face just the first three batters. While the Yankees strategy wasn't "optimal" (i.e., they did not employ {112} as specified by the simulations), it did strike a nice balance between using resources (pitchers), and keeping the opponents' expected scoring low.

Many of the worst possible strategies, labeled at the top of Figure 10, still use Chapman, but only after Pitcher 3 faces the first two batters. It appears that this combination would have been particularly bad for the Yankees, and hence good for the Red Sox.

4.2.2. Boston Red Sox vs. Houston Astros - Game 5 ALCS

Our second case study takes the Boston Red Sox vs. the Houston Astros in the top of the 9th inning of Game 5 of the ALCS, where the Red Sox again led 4-1. The Astros' available pitchers included the pitcher of record after the 8th inning, Roberto Osuna, and he actually pitched the entirety of the 9th inning as well. A selection of five available pitchers and the Red Sox lineup beginning in the 9th inning are shown in Table 5. Figure 11 shows the average number of runs that the Red Sox were expected to score in the 9th inning as a function of pitching strategy, again, highlighting some of the best and worst cases.

Osuna's pitching index is 3 for our simulation, and he actually faced only three batters in the top of the ninth, so we have the same problem as we did for the Yankee's test case. Still, we choose to highlight

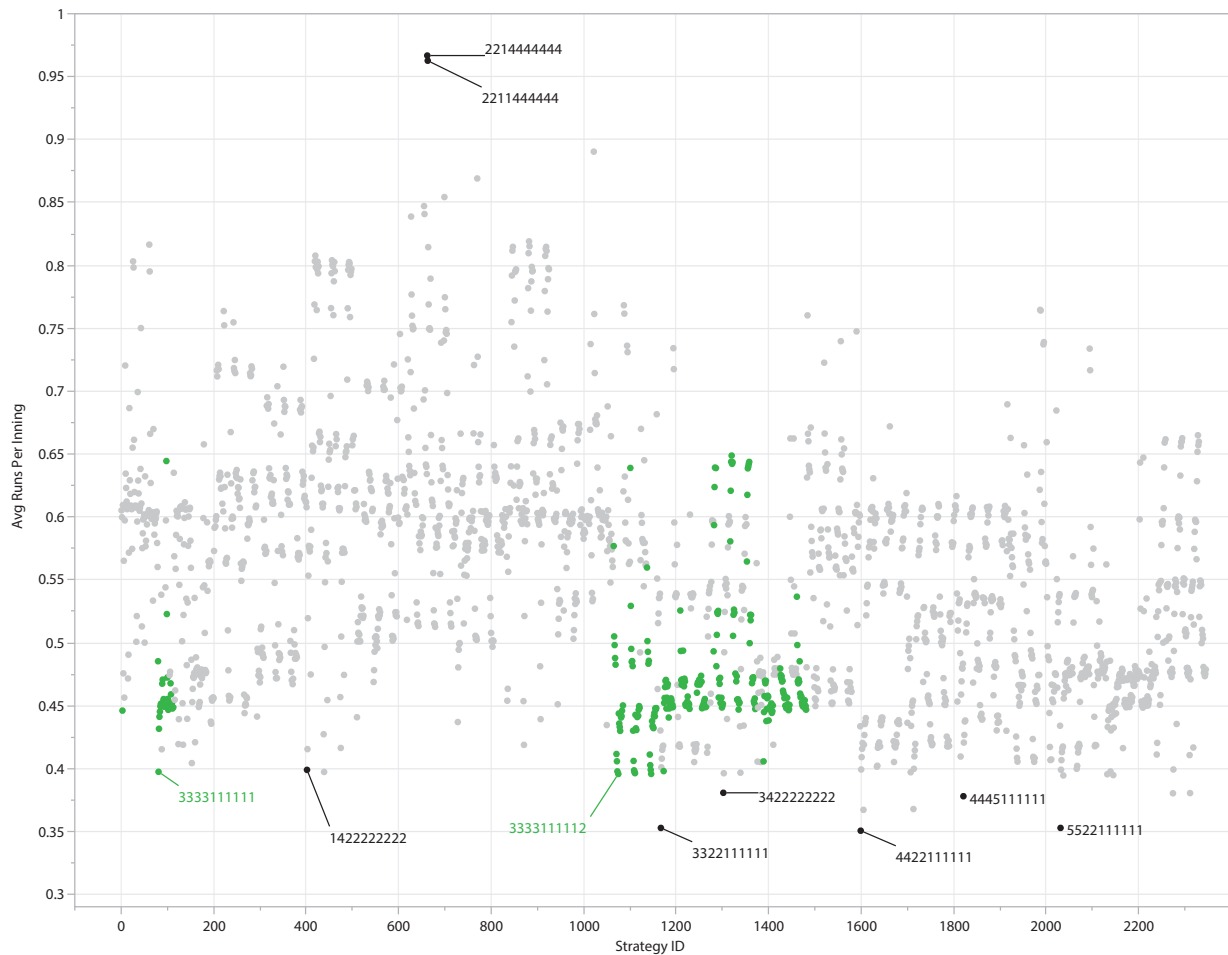


Figure 11: Average number of runs per inning for each pitching strategy with optimal and worst strategies labeled for the final inning of the 2018 American League Championship Series. Labels are for Pitcher ID assignments for 10 batters.

(in green) all strategies consistent with Osuna facing the first three batters in Figure 11. Although none of these were the optimal strategies, we see a number of them that are very good. Hence, we must give the Astros relatively high marks, just as we did for the Yankees.

At this point, we wish to discuss the tradeoff between optimal matchup strategies, and the value of resources (pitchers). While we see other strategies that were slightly better than Osuna facing the first three batters, they are not much better, indicating that a manager may choose to leave Osuna in to conserve other pitchers. If the Astros had been able to tie the game in the bottom of the 9th inning, necessitating extra innings of play, they would have found themselves in a stronger position by not using additional pitchers to get through the 9th inning. Thus we see that a perfectly acceptable approach to using this strategy in real life may be to simply let the manager choose how many resources he is comfortable using to get through any one inning. Simulations can be run with these constraints in place to find a good strategy that strikes the right balance between favorable matchups and the wise utility of pitchers.

5. Conclusion

We have presented a technique for classifying pitcher-batter matchups to provide a set of cluster types, by which all matchups can be sorted. The training requires statistically significant matchup data, and verification of our techniques requires the presence of large bicliques in high pitcher-batter matchup-count bipartite graphs. Upon finding said bicliques, innings of play may be simulated, and various strategies can be compared against the “truth” data (data obtained by using the maximum-likelihood estimates of event probabilities within pitcher-batter matchups for inning simulation). In this paper, we use this approach to verify the utility of selecting 15 cluster types within the matchup data. We then show how these cluster types can be used to simulate potential matchups where data is sparse or nonexistent in the real world. The result is a set of algorithms that can be used to identify optimal pitcher-batter matchup strategies in both cases when matchup data is abundant and when it is sparse. Finally, the 15 Cluster approach outperforms the Lefty-Righty Cluster approach by a significant margin to predict the simulated output for pitcher-batter matchups.

References

- [1] Retrosheet. URL www.retrosheet.org.
- [2] Phil Birnbaum. A guide to sabermetric research. *Society for American Baseball Research (SABR)*. URL <http://sabr.org/sabermetrics>.
- [3] Reinhard Diestel. *Graph Theory*. Graduate Texts in Mathematics. Springer, 3rd edition, 2000.
- [4] Jason Gore, Jeff Price, Sal Syed, Blake Wooster, and David Dusek. Retrosheet, 2017. URL www.sloansportsconference.com/content/golf-taking-new-approach.
- [5] Nobuyoshi Hirotsu and Mike Wright. Modeling a baseball game to optimize pitcher substitution strategies using dynamic programming. *Economics, Management, and Optimization in Sports*, pages 131–161, 2004.

- [6] Nobuyoshi Hirotsu and Mike Wright. Modelling a baseball game to optimise pitcher substitution strategies incorporating handedness of players. *IMA Journal of Management Mathematics*, 16(2): 179–194, Jan. 2005.
- [7] Bill James. *The Bill James Baseball Abstract*. Self-published by Bill James, Mar. 1982.
- [8] Joe H. Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, 1963. doi: 10.1080/01621459.1963.10500845. URL <https://www.tandfonline.com/doi/abs/10.1080/01621459.1963.10500845>.
- [9] Michael Lewis. *Moneyball: The Art of Winning an Unfair Game*. WW Norton & Company, 2004.
- [10] G. R. Lindsey. Statistical data useful for the operation of a baseball team. *Oper. Res.*, 7:197–207, 1959.
- [11] Y. Malgrange. Recherche des sous-matrices premières d’une matrice à coefficients binaires. applications à certains problèmes de graphe. In *Proceedings of the Deuxième Congrès de l’AFCALT*. Paris: Gauthier-Villars, 1962.
- [12] MIT Sloan School of Management. MIT Sloan Sports Analytics Conference, 2017. URL www.sloansportsconference.com/archive/annual/2017-annual-recap.
- [13] Douglas C. Montgomery and George C. Runger. *Applied Statistics and Probability for Engineers*. John Wiley & Sons, Inc., 5th edition, 2011.
- [14] Official Playing Rules Committee. *Official Baseball Rules: 2016 Edition*. Office of the Commissioner of Baseball, 2016. URL http://mlb.mlb.com/mlb/downloads/y2016/official_baseball_rules.pdf.
- [15] Bill Plunkett. Dave Roberts stands by Dodgers’ platoon-system approach. *The Sun*, Oct. 2018. URL www.sbsun.com/2018/10/26/dave-roberts-stands-by-dodgers-platoon-system-approach.
- [16] Yun Zhang, Charles A Phillips, Gary L Rogers, Erich J Baker, Elissa J Chesler, and Michael A Langston. On finding bicliques in bipartite graphs: a novel algorithm and its applications to the integration of diverse biological data types. *BMC Bioinformatics*, 15(110):1–18, 2014. URL www.biomedcentral.com/1471-2105/15/110.