Ilan Goodman (with Elena Frey for CS221)
CS229 Final Report

Beating the Streak: Predicting the MLB Players Most Likely to Get a Hit Each Day

Motivation and Task Definition

Major League Baseball is a statistics-rich sport, with a premium placed on interpreting those statistics and determining their worth. Despite all of the information catalogued over the years of MLB, there is no one clear indicator of the factors that cause a team to win or that lead to a player performing well in a given game. In spite of these challenges, it is possible to discern trends in the data and create a model for the game that applies more accurately than a casual fan might be able to predict from observation.

Joe DiMaggio holds the record for the longest hitting streak in Major League Baseball: 56 games, set in 1941. No other player has come close to that record. In 2001, mlb.com created a fantasy game called Beat the Streak™ in which participants pick a new player each day they think will get a hit. If the player actually gets a hit, the participant extends his or her "hitting streak"; if not, the participant's hitting streak resets to 0. Each day, a participant can choose a new player (or the same player if they prefer), and the first player to reach 57 games wins the grand prize of $5.6 million. This is an unexpectedly challenging task, and no fan has even reached a streak of 50 in the history of the game. With this goal in mind, **we are attempting to beat the streak and predict the batter most likely to get a hit each day using machine learning techniques.** Further, we also wrote a recommendation tool to aid a human player in this competition.

Data

We scraped our data from baseball-reference.com. In order to be able to make predictions across the entire MLB, we trained and tested on data from the last 32 years. We extracted the data from each game played since 1981 and we trained our models on the complete data for batters from the 2008-2012 seasons, and then tested it on the complete data from the 2013 season. We continued to update the machine's weight vector and data with each "day" of testing, as we would when using our program in the competition over the course of a season. To harvest this data we accessed complete transcripts to every game in each season on baseball-reference.com and derived all the data needed from these statistics. We also derived the player data from these games. We downloaded yearly rosters to represent the team. We currently have this database stored, which will come in handy when we test our model in real time in the 2014 season. This database includes data for every batter-pitcher matchup since 1981 and their results, plus auxiliary data about the games such as time and location.

Approach and Features

To implement our machine learning algorithms we developed a feature vector that focused on areas that can have an impact on a batter's likelihood to get a hit on any given day. Features included, but were not limited to, the batters' and pitchers' skills, game conditions, and hot streaks. With this in mind, we chose a set of 50 features to form a 50-dimensional vector representing a single batter for a single game. We scaled each feature so that it falls roughly in the interval [0,1] in order that the learned weights have an easily interpretable meaning. Refer to Appendix A for a complete list of the features. We also have a constant (bias) term in our feature vector, and we mark whether the player gets a hit on each day or not with a 1 or 0. The input for our algorithm is the feature vector for the player and the opposing pitcher, and the output is a measure of how likely that player is to get on that day. We needed to make some simplifying assumptions that undoubtedly affected our training and our predictions due to the lack of data that we could unearth. For example, we assumed that a team's roster (although it can vary daily) was static and enlarged for the entire season, since the daily roster information is not available.

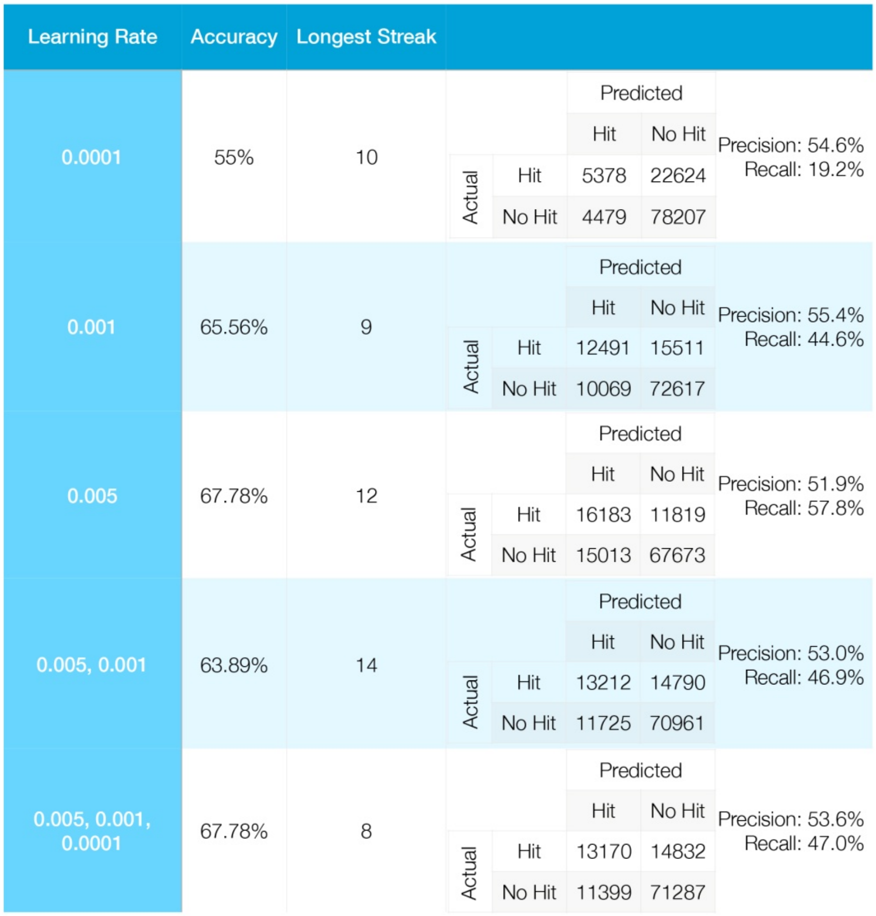
Primary Algorithms

We modeled the likelihood that a player gets a hit on any given day as a generalized linear model (GLM) with a Bernoulli distribution. This is ultimately equivalent to a logistic regression, so we choose to solve our classification problem by implementing logistic regression. Our classifier assigns a likelihood to each player that he will get a hit on any given day (based on his feature vector) via the sigmoid function. Since we have so much data, it suffices to pass over our training data once. When testing and predicting, we selected the player each day who had the highest score (as given by the sigmoid function). In order to create our confusion matrices for analysis, we predicted every player whose score was above 0.5 to get a hit and all the others to go hitless on that day.

Results and Discussion

While our ultimate goal is to predict and maintain the longest possible "hitting streak" of correct predictions, the probability of maintaining such a streak (assuming mutual independence between each pair of days) is slim, even with high prediction accuracy. Accordingly, the best way to measure success is the percentage of times we correctly predicted a player to get a hit. One other way to evaluate our performance is to look at how many players were correctly classified as "hit" or "no hit" on each day.

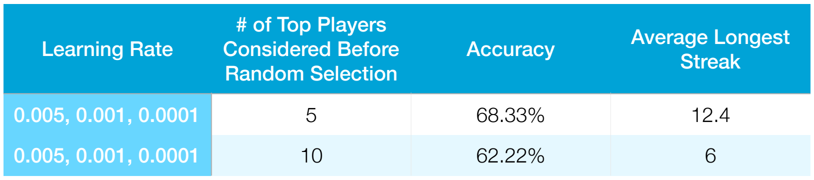
We tested our logistic regression algorithm with several different learning rates and calculated our pick accuracy (the percentage of the time our pick actually got a hit on that day), along with the longest streak we were able to achieve and the confusion matrix for the global predictions we made for each batter each day. When multiple learning rates are shown it means the model was trained once completely with each successive value (e.g., the last row was trained once with a learning rate of 0.005, a second time with 0.001, and a third time with a rate of 0.0001).



Learning Rate	Accuracy	Longest Streak					
0.0001	55%	10			Predicted		Precision: 54.6% Recall: 19.2%
					Hit	No Hit	
			Actual	Hit	5378	22624	
				No Hit	4479	78207	
0.001	65.56%	9			Predicted		Precision: 55.4% Recall: 44.6%
					Hit	No Hit	
			Actual	Hit	12491	15511	
				No Hit	10069	72617	
0.005	67.78%	12			Predicted		Precision: 51.9% Recall: 57.8%
					Hit	No Hit	
			Actual	Hit	16183	11819	
				No Hit	15013	67673	
0.005, 0.001	63.89%	14			Predicted		Precision: 53.0% Recall: 46.9%
					Hit	No Hit	
			Actual	Hit	13212	14790	
				No Hit	11725	70961	
0.005, 0.001, 0.0001	67.78%	8			Predicted		Precision: 53.6% Recall: 47.0%
					Hit	No Hit	
			Actual	Hit	13170	14832	
				No Hit	11399	71287	

The longest streak we maintained was 14 and our peak accuracy was 67.78%. Although these results are significantly above average for a human playing Beat the Streak™ and are better than a baseline measure of picking the player with the highest batting average (62% success rate over a few months of data), these initial results were not as successful as we had originally hoped.

We then modified our classifier to serve as a recommendation tool for a human playing BTS™. Once we made our predictions for each batter's likelihood to get a hit, we selected the top five candidates and to present to the player. To evaluate this system, we modeled the human as a random agent and uniformly chose the batter to get a hit from this list of five. We repeated this process several times and averaged our results from each of these trials. We repeated this with suggesting the top ten players most likely to get hits. These results represent an improvement over our earlier selection algorithm.



Learning Rate	# of Top Players Considered Before Random Selection	Accuracy	Average Longest Streak
0.005, 0.001, 0.0001	5	68.33%	12.4
0.005, 0.001, 0.0001	10	62.22%	6

We found that in 69.5% of days in the first third of the season our recommender provided three or more batters that got a hit from the five total. In the middle third of the season the recommender offered at least three correct batters 85% of the time, and in the last third of the season he recommend was at least 3/5 correct 80% of the time. Together, over the last 5/6 of the season (once the season statistics that comprise the majority of our feature vector are meaningful), the majority of the hitters we recommended got hits 82% of the days. This is a significant result in that it shows how powerful a tool this recommendation system could be for someone playing the game who might have extra information that we could not model for historical data reasons, such as injury history.

Although we did not attain our initial goal of reaching a record-breaking streak, we were able to achieve a prediction percentage that is consistently around 70%. This is still successful as this level of accuracy is well above the human average for the game based on data from Beat The Streak™, and it beats baseline data by 10%. However, even with this above average prediction it is nearly impossible to maintain a hitting streak. Assuming we predict a batter correctly 70% of the time and each prediction is independent, then the chance of us getting merely a 20-game hitting streak is 0.079%, and the chance of a record-breaking 57-game streak would be 1.481x10^{-7}%.

Additional Results

We also approached the problem from another direction: instead of modeling each player-day independently, we tried to model each streak. To that end, we modeled each player's expected actual hitting streak (as well as our own "picked" hitting streak) with a GLM using a geometric distribution. While we could not find any literature to reference for this particular distribution, the geometric distribution falls nicely into the exponential family with the sufficient statistic $T(y) = y$. We find that $\eta = \ln \varphi$ (with probability φ of predicting a player to get a hit on a given day), and since $E[y|x;\theta] = \varphi(1 - \varphi)^{-1}$, we find that we have a hypothesis function $h_\theta = (\exp(-\theta^T x) - 1)^{-1}$, yielding our stochastic gradient update $\theta_{i+1} := \theta_i - \alpha((\exp(-\theta^T x) - 1)^{-1} - y)\, x_i$. We tried to apply our data to this model to see if we could improve our results, but it did not work in our prediction framework because we are predicting the next player to get a hit and not the next longest actual hitting streak. Nevertheless, this is an important novel result because we could use it to model other aspects of baseball that we have not yet incorporated into our features.

Conclusion

While we did not manage to "beat the streak," we did build a program that predicts which major league baseball players are most likely to get hits consistently better than humans and baseline implementations. Even though our theoretical discoveries did not translate into practical code, we built a robust recommendation tool for anyone who wants to play Beat the Streak™.

A tool such as ours could be invaluable to managers and owners of baseball teams. A manager could use this method or something similar to select their daily lineup—even the few percentage points that we would add to the win probability of the game would be worth several million dollars in today's baseball market. Furthermore, we could easily adjust our program to apply for player scouting, suggesting trades, analyzing other sports, and countless other possibilities.

Overall, we wrote a successful artificial intelligence and learned about how machine learning works in the real world, how to obtain and manage large quantities of data, and how our mathematical theory translates to reality.

Appendix A: Feature Vector

- The home run factor for the park the game is played in (how home run friendly this park is; 1.0 is average, below is hard to hit HRs) (0.5-1.5)
- The run factor of the park the game is played in (how easy it is to score a run in this park; 1.0 is average, below is hard to score) (0.5-1.5)
- The hit factor of the park the game is played in (how easy it is to get a hit) (0.5-1.5)
- The double factor of the park the game is played in (how easy it is to hit a double) (0.5-1.5)
- The triple factor of the park the game is played in (how easy it is to hit a triple) (0.5-1.5)
- The walk factor of the park the game is played in (how easy it is to get a walk) (0.5-1.5)
- An indicator if the game is National League (NL) or American League (AL), as determined by the home park (1 or 0)
- An indicator if the game is played during the day or at night (0 or 1)
- An indicator if the game is a home game for the batter or away (0 or 1)
- The time zone shift that the batter is playing in (-3 to 3, discrete)
- The time zone shift that the pitcher is playing in (-3 to 3, discrete)
- The number of hits the player has had in his last game divided by 4.0 (0.0+)
- The number of hits the player has had in the last two games divided by 8.0 (0.0+)
- The number of hits the player has had in the last three games divided by 12.0 (0.0+)
- The number of hits the player has had in the last four games divided by 16.0 (0.0+)
- The number of hits the player has had in the last five games divided by 20.0 (0.0+)
- The number of hits the player has had thus far this season divided by 200.0 (0.0+)
- The batter's season batting average (avg) times 3.3 (0.0-3.3)
- The batter's season on base percentage (OBP) times 2.5 (0.0-2.5)
- The batter's season slugging percentage (SLG) times 2.0 (0.0-2.0)
- The batter's current hitting streak divided by 20.0 (0.0+)
- The batter's season strikeout (K) rate times 4.0 (0.0-4.0)
- The batter's season walk (BB) rate times 5.0 (0.0-5.0)
- The batter's season home run rate times 100.0 (0.0-100.0)
- The batter's season number of at bats per game divided by 5.0 (0.0+)
- The batter's number of years in the majors divided by 20.0 (0.0+)
- The batter's career batting average times 3.3 (0.0-3.3)
- The batter's career batting average in this park times 3.3 (0.0-3.3)
- An indicator if the batter's team won its last game or not (1 or 0)
- The number of hits the batter's team had in its previous game divided by 16.0 (0.0+)
- The head-to-head winning percentage of the batter's team vs. the pitcher's team
- The number of hits the pitcher gave up in his last outing divided by 10.0 (0.0+)
- The number of Ks the pitcher got in his last outing divided by 16.0 (0.0+)
- The starting pitcher's ERA in the past five games divided by 4.0 (0.0+)
- The starting pitcher's season earned run average (ERA) divided by 4.0 (0.0+)
- The starter's season WHIP (walks and hits per inning pitched) (0.0+)
- The starter's season K/IP rate (0.0+)
- The starting pitcher's season winning percentage (0.0-1.0)
- The starter's season K to BB ratio divided by 4.0 (0.0+)
- The starter's average innings pitched (IP) per game divided by 8.0 (0.0+)
- The pitcher's number of years in the majors divided by 20.0 (0.0+)
- The starter's career ERA divided by 4.0 (0.0+)
- The starter's career WHIP (0.0+)
- The starter's career K/IP rate (0.0+)
- The starter's career BB/IP rate times 2.0 (0.0+)
- The pitching team's season bullpen ERA divided by 4.0 (0.0+)
- The pitching team's season bullpen WHIP (0.0+)
- The pitching team's season K/IP rate (0.0+)
- The pitching team's ERA in the past 10 games divided by 4.0 (0.0+)
- An indicator if the pitching team won their previous game (1 or 0)