CS 221 PROJECT PROGRESS REPORT

STUART SY AND YUSHI HOMMA

1. INTRODUCTION OF TASK

ESPN fantasy baseball is a common pastime for many Americans, which, coincidentally, defines a problem whose solution could potentially be predicted by artificial intelligence. The particular ESPN fantasy baseball game that we will analyze in this project is the ESPN Baseball Challenge. The basic task for these fantasy baseball participants is to predict which players will have successful games based on a standard scoring system and optimize the total score of their fantasy team under a budget. The fantasy teams consist of 9 players (one for each fielding position plus DH) and an entire pitching staff. This scoring system is provided by ESPN.com, which we define below in Section 5.

2. LITERATURE REVIEW

Upon searching for literature on similar topics, I came across two related former Stanford CS 229 project papers. "Predicting the Major League Baseball Season", written by R. Jia, C. Wong, and D. Zeng, regarded predicting wins and losses of teams in a particular season. Another that I came across, "Applying Machine Learning to MLB Prediction & Analysis", similarly tried to predict games to be wins or losses based on previous statistics.

3. INPUT/OUTPUT

The input into our system is the previous season's statistics. The system will then use this input to output the optimal team of 9 players of each position and a pitching staff with total budget of $50 million.

4. DATASET

The dataset will include season statistics for every player in the MLB since 1871 matched with the scores of the corresponding players in the next season. We obtain this data from the baseball statistics archive accumulated by Sean Lahman in his collection of statistics found at http://www.seanlahman.com/baseball-archive/statistics/. The statistics for batters consist of the year played, their team, position, age, numbers of games, at-bats, runs, hits, 2B, 3B, home runs, runs batted in, stolen bases, attempts caught stealing, walks, and strikeouts. For pitchers, the dataset includes year played, team, age, salary, number of games, wins, games started, complete games, shutouts, saves, innings pitched (in outs), shits, earned runs, home runs, walks and strikeouts.

Date: November 14, 2015.

5. EVALUATION

The scoring system based on individual stats for the hitters and stats for the entire pitching staff of a team. The score for a single hitter is defined by the following formula:

$$\text{Score}_{\text{hitter}} = \text{R} + \text{TB} + \text{RBI} + \text{BB} + \text{SB}, \tag{1}$$

where $\text{R} = (\text{number of runs})$, $\text{TB} = (\text{total number of bases})$, $\text{BB} = (\text{number of walks})$, $\text{RBI} = (\text{number of runs batted in})$, and $\text{SB} = (\text{number of stolen bases})$.

The score for an entire pitching staff for a season is defined by:

$$3 \cdot \text{IP} - 3 \cdot \text{ER} - \text{H} - \text{BB} + \text{K} + 5 \cdot \text{W}, \tag{2}$$

where IP is the total number of innings piched, ER is the number of earned runs, H is the total number of hits that the pitching staff gave up, $\text{BB} = (\text{number of walkspitched})$, $\text{K} = (\text{number of strikeouts pitched})$, and $W = (\text{number of wins})$.

We will evaluate our system's predictions with the scores on the leaderboard of the ESPN Baseball Challenge listed for the past 4 years. These will be our test datasets.

6. INFRASTRUCTURE

Using the pandas data science library, we extracted relevant parts of the CSV data and processed it into usable python objects. We separated the data by season because each season is one sample of data. We also determined what position each player is categorized as based on the metric given by ESPN.com (having played 20+ games at that position in the previous season). We also matched up the playing statistics in Batting.csv and Pitching.csv with the calculated ages from Master.csv.

7. CHALLENGES AND APPROACH

This project poses two main challenges: to predict the scores of each player/pitching staff for the next season using the previous season's statistics and to find the maximum total team score within the constraints of the budget and player positions.

Since the flow of a baseball game has natural breaks to it, and normally players act individually rather than performing in clusters, the sport lends itself to easy record-keeping and statistics. Ever since the explosion of fantasy sports, the analysis of player/team performance with statistics has become ever more popular/important.

As stated before, our task is to construct the best possible performing team (defined with a custom fantasy baseball score) under a budget constraint. To model this task, we want to model predicting the price/performance of individual players as a linear regression problem, and to model picking the optimal team as either a search or constraint satisfaction problem.

One challenge that may be difficult to completely remedy is that our statistics only include season statistics instead of game-by-game statistics. This means that once we choose a team for a season, we have no additional information to change the members of the team during the season. However, in the ESPN Baseball Challenge, the fantasy baseball participants can change their rosters throughout the season, which could give them an advantage over our system due to fluctuations of players' performance throughout the season. This is why our oracle might not outperform the

top participants of the ESPN Baseball Challenge, but it will be the best that we can aim toward with our algorithm.

8. Baseline and Oracle

We will perform our baseline and oracle on the year 2014. We defined our baseline to be just dividing the budget into equal parts to budget each individual position, and then obtaining the top-scoring player at each position from the previous year. We used 3/10 of the budget on the pitching staff because that is roughly how much MLB teams pay their pitchers. This leaves 3.9 million for each hitter and 15 million for the pitching staff. This resulted in a score of 4583. We defined our oracle to be the top participant in the ESPN Baseball Challenge in the year 2014. The oracle was GEORGETHEBOSS1 who got a score of 8362. Thus, there is room for much improvement.

9. Linear Regression

The primary portion of our task was to predict the each year's scores of each player using the previous year's statistics for that player. There are a number of ways that we could have carried out the task, but we decided on the simple multiple linear regression. An advantage of using linear regression is that there is not much need to know much about how each variable factors into the prediction. Thus, the work lies in finding good features for the predictor to work with.

The features we used clearly needed to include all of the factors taken into account in calculating the score. Thus, for batters, we included the number of runs, hits, doubles, triples, home runs, runs batted in, walks, and stolen bases. For pitchers, we included the number of innings pitched, earned runs, hits, walks, strikeouts and wins. However, we had many more stats at our disposal. We only included the stats that we thought would have a direct effect on the score next year. This included much of the stats regarding their performance in games, and very little of the biographical information like college and hometown.

These other features that we believed to be relevant for batters are: weight, height, strikeouts, times caught stealing, age and salary. We believed that weight, height and age could somehow help predict the longevity of the players' skills and maybe predict the trends of their growth and slumps. We also believed that their salary could help with other unquantifiable factors like star potential or popularity that could factor into their score for the next year. The strikeouts and times caught stealing are negatives for batters that could indicate a potential for slumps; for example, if a fast player is getting caught stealing more often, it could indicate that he will not steal bases as much the next year because his running skills are deteriorating.

For pitchers we had games played, innings pitched (measured in outs), home runs, the opposing batters' batting average, saves, runs, shut outs, earned runs average, weight, height, year and salary. We included weight, height, and salary for the same reasons as the batters. The rest of the game statistics are pretty clear in their potential effects on the next year's score.

For the algorithm, we treated the batters and the pitchers as distinct datasets. We used KFold from sklearn.cross_validation package to cross validate our data in order to avoid overfitting to the training data. We used LinearRegression from sklearn.linear_model package to train a predictor of the scores for each player. We used these predicted scores in the CSP part that is explained in the following section.

10. Constraint Satisfaction Problem

The constraint satisfaction problem involved picking one player from each position and a pitching staff to maximize score with the given budget of $50 million. First we had to sum the salaries and predicted scores of the pitching staff for each team and store them as one entity. Then, we modeled the problem with each position (including pitching staff) as a variable with the domains being all the players with that position. The players were represented by a tuple of their ID, score, salary and position. Then, we only had one factor: the sum of the salaries of each position had to be at most $50 million.

To actually implement the algorithm, we implemented the get_sum_variable function of the CSP homework. To solve the constraint satisfaction problem, we just used the python-constraint package to input the model and it returned all the possible solutions. Then we just took the solution with the maximum score because we know that all these solutions satisfied the constraint.

11. Initial Experimental Results

We ran our algorithm on the 2010 data predicting the 2011 scores. We found out that the linear regression had an average absolute difference from the actual scores of 78.55 for pitchers. We found that it had an average absolute difference of 87.36 for batters. This shows that the choice of features may not have been the best or that the trends of players may not be linear as there is still quite a bit of room for improvement. Unfortunately, the constraint satisfaction problem has shown to be too inefficient to be used on large datasets, so we do not have experimental data results for making the teams. We will try to figure out how to cut down on the problem: perhaps by deleting players whose scores are too low or salaries too high to be considered for the optimal team.