Least Absolute Shrinkage and Selection Operator (LASSO) in Regards to Individual Offensive Baseball Statistics

Tyler Oshiro

May 20, 2013

Contents

1 Introduction

The study of baseball statistics has been around for generations but has only recently begun serious discussion over the past decade. People will argue about whether or not statistics really give any more information than the eye of an expert, but here, I will attempt to use linear regression methods to model batting statistics to overall team wins. More specifically, I will attempt to use a coordinate descent LASSO algorithm to quickly and efficient model this problem. For this problem I will be working with the Lahman's Baseball Database [1].

2 Dataset

As mentioned above, I will be working with the free online Lahman's Baseball Database. Since I am trying to find a linear model that maps batting statistics to overall team wins I have used the batting.csv file contained within the database. I then mapped each individual player's corresponding team and year to the number of wins and the overall percentage of wins per season (partial parsing implementations can be found in utilities.r). I parsed out any player whose statistics had less than 100 at-bats and converted all NaNs to zeroes for the purpose of this study. Furthermore, I removed any depricated fields and non-real-valued features. Each data point represents a single player's statistics for a single year. All of the data preprocessing was done in R and Excel. The final dataset contains 36630 datapoints and 20 features. Table 1 lists what each feature in the dataset represents.

3 About

Since we want to perform "feature selection" for our data (i.e. choose which statistics of a baseball player are important for winning games), we will use LASSO to reduce the number of possible dimensions down from twenty to approximately ten. The regularization penality that we use for LASSO is that of the L_1 norm. Doing so guarantees sparse solutions. We will also vary this sparsity of our resulting weight vector so long as we can compute it reasonably efficiently.

4 Logistics

Unfortunately, we cannot differentiate our objective function (1) easily and there is no closed-form solution for the minimization so we resort to an iterative coordinate descent method. We can do so because the LASSO objective is convex. Thus, using the subgradient will result in an optimal result.

Let

N = the number of datapoints we are processing.

k = the number of features for the data.

λ = the regularization constant.

$t(x_j)$ = the response variable given a datapoint x_j.

w_0 = our intercept for the optimization function.

w_i = our weight vector for the data.

$h_i(x_j)$ = the weight of feature i for datapoint x_j.

Then the objective function we are trying to maximize is given by:

$$\hat{w} = \min_w \sum_{j=1}^{N} \left(t\left(x_j\right) - (w_0 + \sum_{i=1}^{k} w_i h_i(x_j)) \right)^2 + \lambda \sum_{i=1}^{k} |w_i| \quad (1)$$

5 Regularization

Regularization is done to penalize "complex" solutions with large weights. To choose our regularization constant we will use k-folds cross-validation to try to limit our chance of overfitting our solution to our training data. That is, we will split the data into k (in this case 10) evenly distributed subgroups and then train our algorithm on all but one of the k subgroups and then test our results with the remaining subgroup. We repeat this process k times and average the errors reported. The final λ value will be chosen based on a combination of producing the lowest error and most reasonable runtime using this technique. The k-folds error that we compute is just the sum of the squared errors on the testing data divided by the number of testing samples.

6 Results

For varying regularization parameters ranging from 0 to 1,000,000, the runtime of the algorithm was approximately a few seconds (none above 15 seconds). Note that these results were on the Sage Cloud server. What this means is that we do not necessarily have to do any dimensionality reduction due to the inate linear correlation within the data that causes it to converge quickly. As a sidenote, using large regulariztion constants show that the most important features to determining wins are the number of games played, at-bats, and the number of runs scored which logically makes sense (the more chances you get to play since you are a regular player and the more times you score, the higher the chance that your win a game). For regularization constants of 0 (which yield the lowest k-folds error), we get errors of approximately 0.0073. This means that

the average error for each testing sample was only 0.73%, which, considering a 162 game season (although it is lower in some years), is only an error of 1.18 wins per season. This number is pretty good for predicting overall team wins. It may be noted though that the output considers only a single player's offensive statistics and can greatly skew the overall output of a team of 25 players.
I also applied my results to what I might consider an "average" player for the 2013 Seattle Mariners as of May 30, 2013. I chose Michael Saunders [2]. We are currently about 1/3rd of the way through the season so I multipled all the statistics by a factor of 3. For the purpose of this test I used a regularization parameter of 10000 and a convergence tolerance of 0.001 to produce my weight vector. Using these numbers I got a win probability of 0.476 which corresponds to about 77 wins for the season. It will be interesting to see if this holds true as the over/under betting line for Mariners wins as of February 2013 was 76.5 wins. More results can be seen in project.sagews.

Table 1: Meaning of the Data

Column	Statistic#1	Description#2
1	G	Total number of games played
2	G_batting	Total number of games played where the player had at least one at-bat
3	AB	Total number of at-bats
4	R	Total number of runs
5	H	Total number of hits
6	X2B	Total number of doubles
7	X3B	Total number of tripless
8	HR	Total number of homeruns
9	RBI	Total number of runs batted in
10	SB	Total number of stolen bases
11	CS	Total number of time caught stealing
12	BB	Total number of walks
13	SO	Total number of strikeouts
14	IBB	Total number of intentional walks
15	HBP	Total number of times hit by a pitch
16	SH	Total number of sacrifice hits
17	SF	Total number of sacrifice flies
18	GIDP	Total number of times grounding into a double play
19	Wins	Total number of team wins for the season
20	Win_P	Team winning percentage for the season

References

[1] Sean Lahman, "Lahman's Baseball Database." Sean Lahman. N.p., n.d. Web. 30 May 2013. <http://seanlahman.com/baseball-archive/statistics>.

[2] "Michael Saunders." Baseball Reference. N.p., 30 May 2013. Web. 30 May 2013. <http://www.baseball-reference.com/players/s/saundmi01.shtml>.