Leopold Ringmayr
10/30/2019
COMP 347

COMP347 Project 2: Why Los Angeles Should Adopt a Bike-Share Program

For this project, the goal is to use data on bike-share program usage in Washington D.C. to build a linear model that could predict the demand of a similar program if it were to be implemented in Los Angeles. With this, the aim is to leverage the model to argue in favor or against a bike-share implementation in Los Angeles. For this, I select important features from the Washington dataset to include in the model, perform linear regression using gradient descent, and compare this gradient descent approach to MATLAB's built-in regression capabilities. Linear regression via gradient descent is used to predict the demand, measured in number of bike-share users, for bike-share program. Initially, the dataset included 15 features.

Feature Selection:

For feature selection, three methods were primarily employed: eliminating features by inspection, i.e. domain knowledge, examining correlations between features, and using lasso regression to help identify whether features are irrelevant. Before running correlation tests and the lasso regression, we can identify columns that most likely are not useful to include in our gradient descent regression model. The first approach was to examine the features in the dataset and to determine which subset of features appeared redundant or unhelpful for a model geared towards Los Angeles. We see by inspection that the instant column does not necessarily provide any insight, as this feature resembles a count in steps of one. In addition, we notice that the date column can omitted in favor of leaving in other time-indicating features, namely the year and the month. Taking out holiday also makes sense, seeing as the number of holidays are very limited, and are on very specific days. Hence, this feature does not shed any significant light on bike-share usage on a more generalized scale; the less correlated feature that was left in the dataset was "workday". To highlight the demand on regular workdays such as commuting, the workday feature would be important to include. The two columns temp and atemp are very similar in values, and our correlation test confirmed this by eliminating the temp feature. Because these two measures of temperature are very similar, it makes sense to select only one for the gradient descent portion of the analysis. The casual and registered features are contained within our response variable, the total count. Hence, these two features were omitted as well. I tested my remaining features on correlation and ran a lasso regression with them to find whether there were unwanted correlations between them, or whether any of the features were irrelevant (via lasso regression). Lasso regression allows us to see which coefficients shrink to zero quickly and are irrelevant; for this, the parameter lambda is important. Using cross validation on the training set, one can find optimal lambda. In addition, I chose a lambda based on its index in the FitInfo struct returned by lasso in MATLAB. By seeing at which index of lasso's B a feature is first eliminated, we can establish which lambda to choose. After running lasso in the cross validation, I found a lambda of approximately 0.7 to be optimal. I ran lasso on

the whole training set again using this lambda to find possible irrelevant features. This revealed that none of the six chosen features were irrelevant, i.e. reduced to 0. Choosing feautures entailed finding features that were not correlated or irrelevant, and that would be useful for model predicting the bike-share demand in a city such as LA (features related to climate would be important).

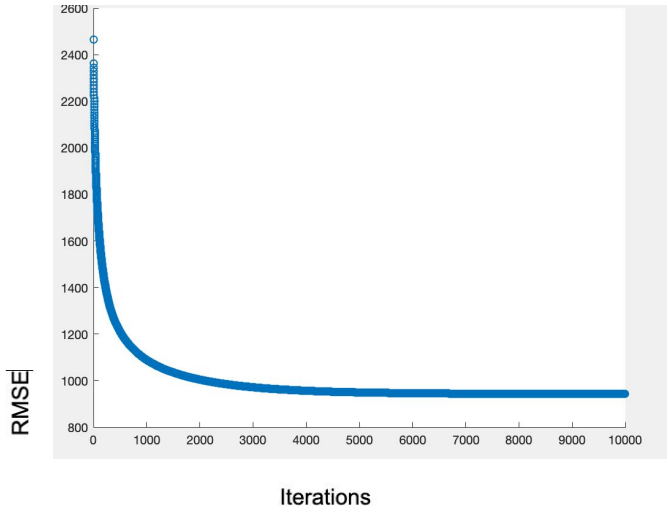
Ultimately, I left six features in my dataset after feature selection–ones that were not correlated and not deemed as irrelevant after lasso regression, which were used for gradient descent. The features I used for the regression via gradient descent are: “year”, “month”, “workday”, “atemp”, “weather situation”, and “windspeed”. My response variable was the total count of users.

Gradient Descent:

The six features were used for the gradient descent process. For this, I found the partial derivatives of the Mean Square Loss function with respect to each of the features. I ran the gradient descent for 10,000 iterations at a learning rate of 0.01.

Results:

While running my gradient descent, I stored my coefficients in order to calculate residuals, and ultimately, the Root Mean Squared Error (RMSE) at each iteration of the gradient descent. My findings showed a convergence of the RMSE; the graph below shows the RSME at each iteration. One can see how the error converges, suggesting that the coefficients are not changing.



Using the built-in *fitlm* regression in MATLAB, one can see that the coefficient from *fitlm* are very similar those derived through gradient descent, thus suggesting that the coefficients have the

correctsize for the model. The fitlm output is pictured below. It is notable that the rows in my data are randomized during the correlation test; therefore, values may vary. Overall, this should not impact the results significantly.

```
ans =
Linear regression model:
    y ~ 1 + x1 + x2 + x3 + x4 + x5 + x6

Estimated Coefficients:
                   Estimate      SE       tStat       pValue
                   ________    ______    ________    __________

    (Intercept)     1288.7     227.15     5.6733    2.2158e-08
    x1              2092.2     80.827     25.885    1.1105e-98
    x2              80.117      12.19     6.5721     1.109e-10
    x3               183.9     86.679     2.1216      0.034294
    x4             -722.25     75.567    -9.5578    3.4031e-20
    x5              6240.2     256.73     24.307    1.9302e-90
    x6             -2356.6      538.7    -4.3745    1.4437e-05


Number of observations: 585, Error degrees of freedom: 578
Root Mean Squared Error: 974
R-squared: 0.758,  Adjusted R-Squared: 0.755
F-statistic vs. constant model: 301, p-value = 3.48e-174
```

Coefficients from gradient descent

beta0	11706
beta1	20582
beta2	88.2889
beta3	198.4
beta4	-763.4
beta5	62575
beta6	-1633

The coefficents from gradient descent fall within the standard errors of fitlm's coefficients. In addition, fitlm's output shows a very low p-value for these findings. Once convergence was achieved, the model could be used to predict the number of bike-share users in Los Angeles. Based on this, I used data I found typical for LA, medium windspeed, high temperature, on a working day in spring. For several of my inputs, the predicted value was well over 1,000. This

speaks in favor a bike-share program in LA; however, one would need to adjust this results to account for the larger population.

Conclusion and Future Work:

Overall, the linear model containing coefficients for the six features suggest that there is a high demand for a bike-shar program in LA. For future work, one could re-evaluate which features are necessary. For example, one could perhaps leave out the season feature; after all, seasons in Los Angeles are very different compared to those in Washington D.C. Therefore, using seasons to argue for a high bike-share demand may not be the most productive approach. For a city like Los Angeles, it was important to include measures of weather and climate in the model; these factors are likely to influence the involvement of Los Angeles citizens in a bike-share program. A feature that could have been useful to include would be a measure for population or density of population. Given LA's large area and large number of residents, especially compared to Washington D.C., the number of people in certain areas of the city are an important indicator whether a bike-share program should be implemented. Overall, Los Angeles should adopt a program similar to that of Washington, as my model shows, the number of users would be very high when testing the model on Los Angeles data. However, it unclear whether there is overfitting in the model, and whether the chosen features are an accurate predictor for the actual bike-share demand. Other features, that are specific to LA, could drastically change the regression model.