Final Project - Basic Probability: Programming

Sven Cornets de Groot
ILLC - Faculty of Science
University of Amsterdam
Science Park 107
1098 XG Amsterdam

Giuliano Rosella
ILLC - Faculty of Science
University of Amsterdam
Science Park 107
1098 XG Amsterdam

Abstract

In this report, we present a linear regression model (LRM) we implemented for the Boston Housing Dataset, to analyse the relationship between features and the corresponding targets. First, we will provide a formal description of LRM and how we implemented it. Then, we will go through the results we obtained in our experiments, ending in our best results after scaling and erasing the raw data.

1 Introduction

The Boston Housing Dataset(1) is a dataset containing data of 506 cities in the Boston district (cases). Each case of the dataset has 13 features and one target value (the median house price of the city). The aim of our analysis is to predict the target, using linear regression.

Denote y_i as the actual value of the median price of city i, $\hat{y}_i$ the predicted value and x^i_j the j^{th} feature of case i. Linear regression uses the following prediction function: $\hat{y} = \theta_0 + \theta_1 x_1 + ... + \theta_n x_n$, with n the number of features taken into consideration. The goal is to find $\theta = (\theta_0, \ldots, \theta_n)$ such that the cost function $J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2$ is minimized, with m the number of cases in our dataset ($m = 506$).

The accuracy of the prediction function depends on the values of the parameters θ_i: the more accurate the parameters are, the more the predicted prices will be closer to the actual prices in our dataset. The cost function measures this accuracy: it takes the parameters as arguments and returns the mean of the squared distance between the predicted price and the actual price.

Finding the best parameters amounts to finding the minimum of $J(\theta)$, using gradient descent (GD). Starting with initial parameters $\theta = 0^n$, in each iteration of GD we update our parameters simultaneously: $\theta'_j = \theta_j - \alpha \cdot \frac{\delta}{\delta\theta_j}J(\theta)$ for every j. Here, $\frac{\delta}{\delta\theta_j}J(\theta)$ is a partial derivative of J and α is intuitively understood as the step-size in the iteration of GD.

At each step of GD new parameters, approaching the minimum of J, are returned; the aim of this process is to obtain the best values which minimize J. Hence we set our model in such a way that it performs GD until the difference between the values of $J(\theta)$ at iteration $n-1$ and at iteration n is smaller than 0.0001 (which is approximately at the minimum). After finding such parameters, we can calculate the prediction $\hat{y}$ using them.

First we programmed a Baseline LRM, using no features. There we see that $\hat{y} = \mu$ with μ the mean of the actual values of the prices. Then, we started analyzing the features in the dataset trying to make predictions on each of them and comparing our results with the prices in the target. A measure for this comparising is $R^2 = 1 - \frac{\sum_{i=1}^{m}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{m}(y_i - \frac{1}{m}\sum_{i=1}^{n} y_i)^2}$. The closer R^2 is to 1, the better our result. At last, we tried to make improvements on our dataset, in order to find better predictions using LR.

2 Improvements

In this section, we will focus on the theoretical description of the improvements we implemented in our functions. They are:

(1) scaling of the features;

(2) powering;

(3) erasing of corrupted values.

(1) amounts to restricting the range of the features in order to make GD more efficient and accurate: we implemented a function such that it replaces each value x_i in the initial matrix of features with $\frac{x^{(i)}_j - \mu_i}{\sigma_i}$, where μ_i is the mean of the values of the i-th feature and σ_i the standard deviation.

(2) is a specific method to improve the single variable linear regression: it modifies the prediction function into a polynomial of degree n; more formally, from $h(x) = \theta_0 + \theta_i x_i$ it returns $h(x) = \theta_0 + \theta_1 x_i + \theta_2 x_i^2 + ... + \theta_n x_i^n$.

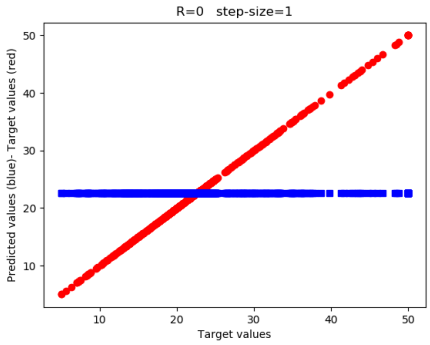
(3) modifies the target matrix by deleting the cases withvalue 50 and modifies the initial matrix by deleting the cases, i. e. the rows, corresponding to these target values. As 50 is the maximum target price, occuring 16 times, we suspect that these values are corrupted. Disregarding them may lead to better results.

By running experiments we can in practice test how these improvements affect our linear regression.

3 Experiments

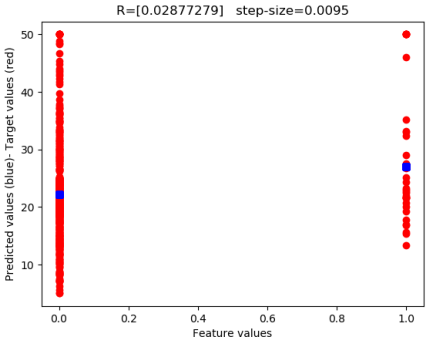
In this section, we will show the results of the experiments we ran and try to make sense of them in order to get information about the features in the dataset.

The first step of the linear regression was to implement a baseline model (BM), which consists of a prediction of the prices when no argument, i. e. no features, are given. Hence we obtained the following:



where the blue dots are the predicted values and the red dots are the values in the target. As we can notice, the BM just returns the mean of the prices in the target as no other feature is given, hence its R^2 is 0, which is a bad result. Then, we started analysing the features in the dataset; the first property we noticed was that all of them are continuous, namely they take values in the continuous space, except for the categorical feature 3 [1] which takes either 0 or 1 as values. In fact, predictions under such feature look like

[1] Notice that we indexed the features from 0 to 12.



After that, we implemented a single variable linear regression model for each feature; the aim of that was to understand how the features individually contribute to making good predictions. Hence, we obtained the following outputs

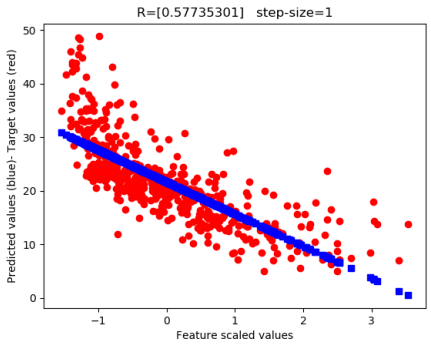
Features	R^2	**Step-size**
0	0.148799	0.02
1	0.129289	0.002
2	0.233559	0.01
3	0.028772	0.0095
4	0.182529	1.5
5	0.481064	0.04
6	-1.49333	0.000005
7	0.062415	0.1
8	0.145378	0.01
9	-1.41016	0.000005
10	0.082475	0.0005
11	0.030250	0.0000005
12	0.534292	0.0005

where the **Step-size** is the value of α in the gradient descent. We can notice that the prediction under feature 12, i. e. the lower status of the population (LSTAT), has the highest R^2; this means that the predictions based on LSTAT are in general more reliable than the ones based on another feature. Now, we tested the improvements on the linear regression model of LSTAT and obtained the following results:

Feature	**Improvement**	R^2
12-LSTAT	none	0.5324292
12-LSTAT	Scaling	0.5441463
12-LSTAT	Erasing	0.56326049
12-LSTAT	Scaling+Erasing	0.57735301

After erasing the corrupted values and scaling the features, we obtained a satisfying approximation of the predictions to the actual prices as we

can see from the graph:

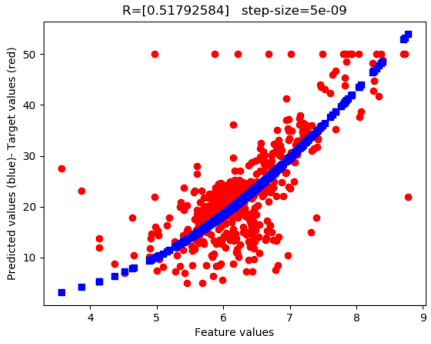


.

We also tested the powering improvement on the feature 5, i. e. average number of rooms per dwelling; we can see from the initial table that the predictions under RM have a similar degree of accuracy to LSTAT. Then we obtained the following results:

Feature	**Improvement**	R^2
5-RM	none	0.481064
5-RM	powering	0.51792584

and the following graph:

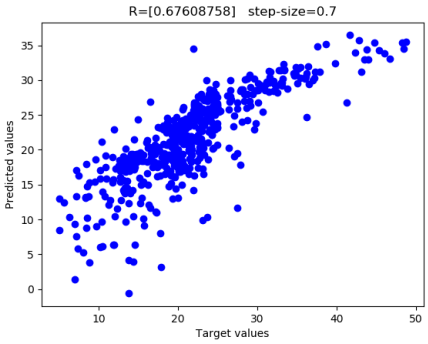


We can notice that by the powering improvement we obtain a polynomial prediction function whose degree of approximation to the actual prices is better that the linear prediction without any improvement.

At the end, we implemented multiple variable regression models; firstly we took in consideration the three most reliable features according to the initial table, namely 2, 5 and 12 and we improved the initial prediction based on them:

Features	**Improvement**	R^2
2, 5, 12	none	0.63977322
2, 5, 12	Erasing	0.675069
2, 5, 12	Scaling	0.63998457
2, 5, 12	Erasing+Scaling	0.67608758

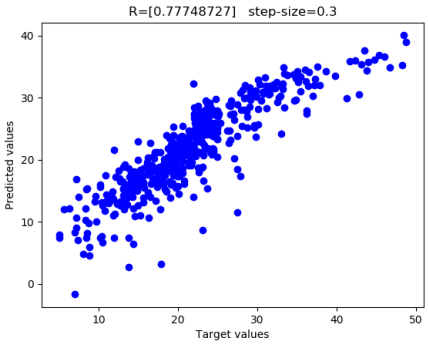
ending in the following graph:



Then, in order to get a better R^2 error we took in consideration all the thirteen features in the dataset and we obtained the following table:

Feature	**Improvement**	R^2
all	Scaling	0.73906227
all	Erasing	0.74051706
all	Scaling+Erasing	0.77748727

$R^2 \sim 0.777$ was the best degree of accuracy we got; hence the predictions under the thirteen features should approximate very well the actual prices in the target; the graph gives also a visual proof of this fact:



The values on the x-axis stand for the predicted prices while the values on the y-axis are the actual prices in the target; the blue dots represent the relationship between the predictions and the target: intuitively the more accurate our predictions are, the more compact and linear is the distribution of the dots in the plot.

4 Conclusions

Our analysis was set for descriptive purposes: we wanted to understand better the features in the dataset and how they are related to the predictions. From the results of the single variable regression models, it turned out that features 2, 5, 12 are more reliable than the other ones to get good predictions.

On these models, we tested the implemented improvements: it turned out that erasing corrupted values increases the degree of accuracy of our predictions better that the scaling method. The powering method also proved to have a good degree of improvement.

Our final multivariable regression model, based on all the features, turned out to have a satisfying degree of accuracy and it guarantees to obtain in general reliable results of prediction.

In conclusion, we can indicate further improvements, which could be interesting for our model:

(i) *Regularization*: in order to face the problem of overfitting with the power method we could implement a regularization method for high order polynomial functions.

(ii) *Correlation*: in order to get more accurate descriptions of the features and their relationship with the prediction, it could be interesting to analyse the correlation they have with each other and with the target value. On approximate prevision, we could deduce that the features with the highest degree of reliability, i.e. LSTAT and RM, may be more correlated to the actual prices than the other ones.

5 References

(1) *The Boston Housing Dataset*, 1996, URL: http://www.cs.toronto.edu/ delve/data/boston/bostonDetail.html, accessed on 1-06-2018.

(2) *Machine Learning*, URL: https://www.coursera.org/learn/machine-learning/home/welcome.