Brute Force Search for Simple Linear Regression Machine Learning Homework1

Erdoğan Abacı
150315025

erdoganabaci97@gmail.com

Computer Science Engineering
Celal Bayar University – Manisa/Turkey

March 10, 2019

Keywords: LotArea, Price Index, Houses Dataset,
House price estimation, Brute force algorithm

Abstract

Most of the available automatic home price estimation systems makes a number of calculations considering each variable. In this article we will try to estimate the prices that variable by square meters. I will make price estimation with the rough force and decide on how much it will be in the future. I'm going to visualize the encoding and data of the tool I use with python. The data set was taken from the kaggle web platform.

1 Introduction

Constructing a price index for residential properties involves the well-known challenges of large heterogeneity among properties, and sparse transaction data for particular properties. The brute force approach aims to circumvent these limitations allowing the comparison between houses by attributing prices to their individual observable characteristics. In this paper, we implement a estimation for these prices using the simple linear regression, which has been receiving a lot of attention in the fast-growing statistical learning literature.

Essentially, this approach imputes prices for individual dwellings by learning the relationship between observed prices and characteristics from a representative sample of houses in a very exible, non-parametric form. A large number of simple linear regression is estimated sequentially for the original data set, and the resulting residuals obtained by comparing observed and predicted values, based on a set of observable characteristics. Finally, each simple linear regression allows us to summarize and study relationships between two continuous (quantitative) variables: One variable, denoted lot area, is regarded as the predictor, explanatory, or independent variable.The other variable, denoted prices, is regarded as the response, outcome, or dependent variable.

Imputation of these predictions for the observed characteristics of dwellings in die rent points in time results in the necessary information for the chosen price-index linear regression formula. Information on both prices and characteristics are provided by a large data set of appraisals. Selecting random values is very important for machine learning algorithms. Randomly selected points allow the algorithm to predict healthier. Therefore, our aim is to select random values from train and test files and draw the best prediction graph.

Each data is very important to determine the graph which we choose. We are drawing the graph with the brute force algorithm to find the smallest result in the range of specific slope and y values. It can take a long time to predict because we will find out the data because we will find the contrary data. Our purpose is to find the minimum value and minimize the distance between the data and the graph

2 RSS and Brute Force Algorithm

The residual sum of squares (RSS), also known as the sum of squared residuals (SSR) or the sum of squared errors of prediction (SSE), is the sum of the squares of residuals (deviations predicted from actual empirical values of data). It is a measure of the discrepancy between the data and an estimation model. A small RSS indicates a tight fit of the model to the data. It is used as an optimality criterion in parameter selection and model selection.

$$RSS = \sum_{i=1}^{n}(\varepsilon_i)^2 = \sum_{i=1}^{n}(y_i - (\alpha + \beta x_i))^2$$

Our purpose is calculate RSS for each Simple Linear Regression(SLR) line of (W0.W1) pair. Record the RSS values with (W0.W1) coefficients. The minimum value of RSS shows us our (W0.W1) pair. So.we choose these W0.W1 optimum value.After :we determining the SLR Line. feed test set (460 records).We predict the SalePrice of each 460 house using their LotArea. In this case. the algorithm which we use to find the rss values with the brute force algorithm.

3 The Dataset

While data on actual house prices is the ideal information for calculating a residential price index, it is not realistically available on a regular frequency for situation of home. As these assessments will change for each home, most houses are priced at a specific square meter. It is almost impossible to estimate the LotArea at the given outliers because we have trained the data according to the concentrating points. **If we want to achieve healthier results, we should ignore the given outlier data's. In addition, I diverted the training data and test data into pieces randomly, and paid attention to the irregular distribution. I blocked a possible false estimate.**

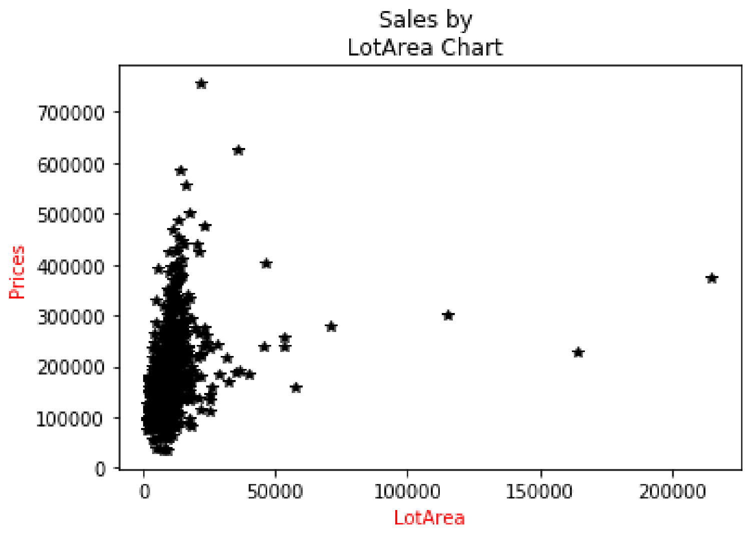


Figure 1: Sales by LotArea Chart

4 Results

The estimation process uses cross-validation within the sample for tuning parameters of the simple linear regression, which allows for an ideal compromise between minimizing mean squared predictions errors and overtting..

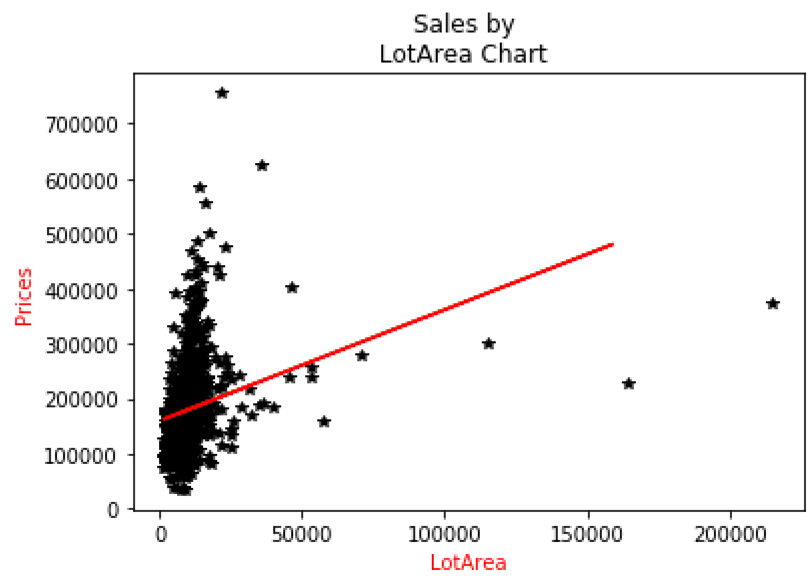


Figure 2 Sales by LotArea Chart with Linear Regression Line

The results show good error, with an out-of-sample root mean squared error within 1000 units of actual (appraisal) prices. Some outliers can be seen in the residuals, which are checked for measurement and coding errors, and eventually include from the dataset for the calculation.

The relative importance of each covariate in the model for reducing total squared errors is best seen on the graph below. Reduction of the Rss value is proof that it draws the graph with the best results.

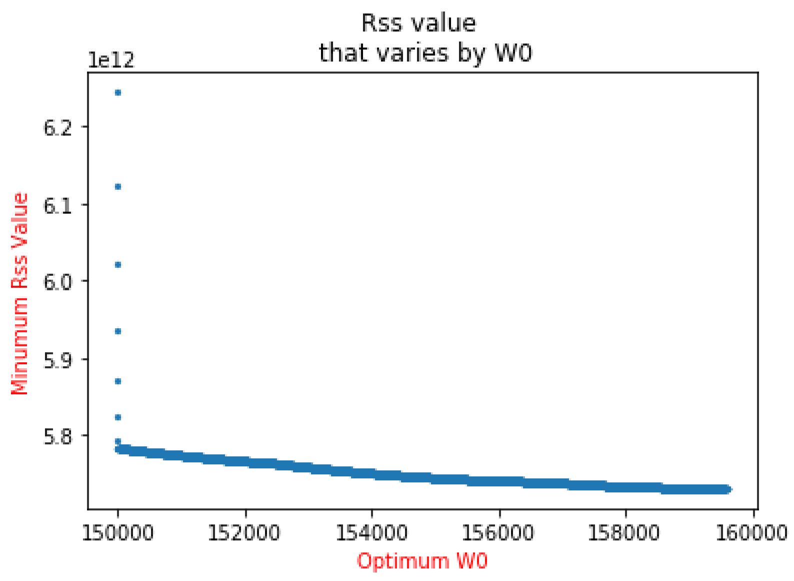
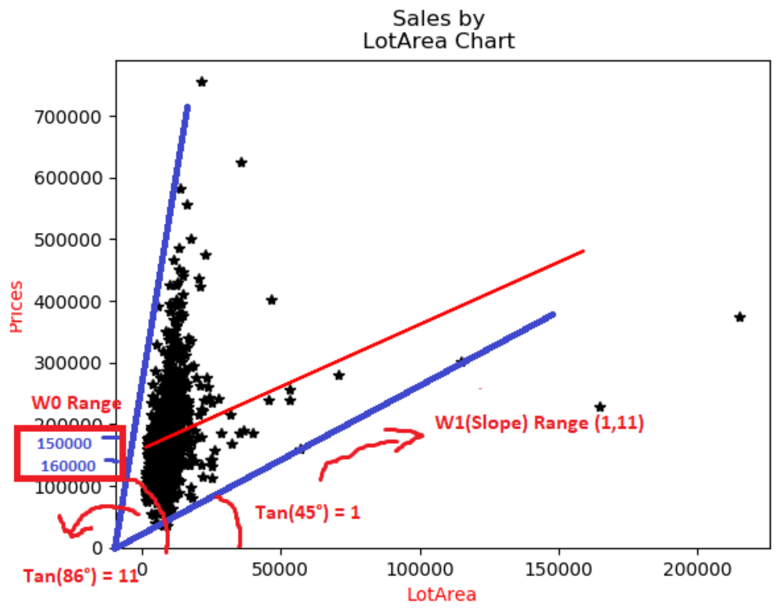


Figure 3 Rss value that changing by W0

Initially, the Rss value drops too fast to find where the graph will be.If he approaches the min point, the line is on a regular basis.

The ranges of w0 and w1, which we selected the RSS, also play a big role. The price values are between 150 thousand and 170 thousand. So if we give the range between 150-170 thousand w0 value.We can find the closest result.

If we want to find the slope of the chart, we consider that the is the closest to the angle values.Which means that according to this data's slope range is Tan(45,86) => (1,11).It is the closest result for Rss value.



5 Conclusion and Discussion

Prediction accuracy for house prices using home Lot Area data along with their intrinsic characteristics in a exible manner just as considering machine-learning algorithms such as simple linear regression boosting for constructing residential price indexes. If the values in the range w0 to w1 are close to the data set, we will increase the performance in our algorithm.The higher the frequency between the received data, the easier it is to predict. But the data given to us was contrary to the data. If we ignore the given data, we could get better results. The Brute force algorithm tried to find the best Rss value for all data. Due to the outliers, we've reached the late result.If we ignore outliers ,brute force algorithm get better solution to find Rss value and drawing graph,predict good values.

6 Code Explanation

Firstly; I wrote the code with python and my compiler is spyder.I Imported 3 library. Matplotlib Allows us to drawing data.
import matplotlib.pyplot as plt

Numpy Library allows to convert series to array
import numpy as np

Pandas library to open csv and seperate column which we want it.
import pandas as pd

This code provide seperate column which we want LotArea and SalePrice
lotarea = data["LotArea"]
prices = data["SalePrice"]
Sklearn library provide separate train and test file randomly to get better solution.
from sklearn.model_selection import train_test_split

We Divide Lot Area and Prices randomly.
x_train,x_test,y_train,y_test=train_test_split(lotarea,prices,test_size =0.31499,random_state=0)

Randomly sort the received values.
x_train=x_train.sort_index()
y_train=y_train.sort_index()

Draw the taken and separated data.Acoording to LotArea and prices.
plt.plot(x_train,y_train,"*",color='k')
plt.xlabel("LotArea",color="r")
plt.ylabel("Prices",color="r")
plt.title("Sales by \n LotArea Chart ")

w0_range = (+150000,+160000) Y axis values naked eye desicion comes values range between +150 000 with +160 000 increased and we don’t want to change list values so we use tupple.
w0_range = (150000,160000)

Graph slop's change between tanjant 86 with 45. We can't give tanjant degree inside range to keep fast solution we give tanjant degree values tanjant86=11 tanjan45 = 1 .We use tupple,Because we don't want to chage type.
w1_range = (1,11)

Divide the default by 50 equal intervals and increase the range by 50 equal parts.Give the initial value each of them.
w1_range_array = np.linspace(w1_range[0],w1_range[1])

We convert to serie to array int to use for loop.
x_train_np = np.array(x_train)
y_train_np = np.array(y_train)
x_test_np = np.array(x_test)

We start to infinite variable and we are constantly bringing the result closer to the minimum
minimum_rss = float("inf")

With Brute force step by step trying to find min Rss value.
for w0 in range(w0_range[0],w0_range[1]):

The following loop calculates the rss value for each w0 and w1 values.For all x values min is multiplying the rss values.
for w1 in w1_range_array:

Multiply the data in the multiplication by 2 and print the each 1000' Rss value. Like ax+b assign data .We draw the graph and repeat for each value.
estimated = (w0+(w1*x_train_np))

Y'estimated - y real value remove and we sum the squares.
rss = sum((estimated - y_train_np)2)**

We find the minimum Rss we assign the minumum w0 and w1 because we find the min rss value and optimum line graph.

```
if rss < minimum_rss:
        minimum_rss = rss
        best_w0 = w0
        best_w1 = w1
        count+=1
        rss_array.append(minimum_rss)
        best_w0_array.append(best_w0)
```

Each 1000 value print the count,With every 1000, print the remaining 0 from the section. I mean print in a thousand if,rss reach

```
 if count % 1000 == 0:
          print("{}.Rss value : {} , W1(slope) value: {} , W0(y column) values: {}".format(count,rss,best_w1,best_w0))
```

Outside the loop is the best result.
print("Result Min W1: {} ,Min W0: {}".format(best_w1,best_w0))

The LotArea values to be estimated are given to the regression graph that we estimate with a cycle and we make a price estimate.

```
 y_predict = []
y_predict_var = 0
for predict in x_test_np:
   y_predict_var =  ((best_w1 * predict) + best_w0)
   y_predict.append(y_predict_var)
print("Predict Values :",y_predict)
```

First, we draw a regression graph which we estimate with the data set.

```
plt.plot(x_test_np,y_predict,color="r")
plt.show()
```

We draw w0 values with Rss.

```
plt.scatter(best_w0_array,rss_array,s=5)
plt.xlabel("Optimum W0",color="r")
plt.ylabel("Minumum Rss Value",color="r")
plt.title("Rss value \n that varies by W0")
plt.show()
```

References

Medium.(2019).MediumWebSite:
https://medium.com/@lachlanmiller_52885/understanding-and-calculating-the-cost
function-for-linear-regression-39b8a3519fcb Received from

Mmsrn. (2019). Mmsrn Web Site:
http://www.mmsrn.com/tum-acilarin-trigonometrik-oranlari-degerler-tablosu/
Received from

Wikipedia.(2019).WikipediaWebSite:
https://en.wikipedia.org/wiki/Residual_sum_of_squares Received from

Youtube. Youtube UdacityWebSite:
https://www.youtube.com/watch?v=E1XzT619Eug Received from