CSE4020

Machine Learning
WINTER SEMESTER 2018

INSTRUCTOR: Prof. Premalatha

STUDENT: Aman Saha [15BCE1273]

Lab Assignment - I

A Study of Linear Regression on Boston Housing Dataset

Abstract

A real estate forecast model is proposed to predict the right price, which is based on linear regression prediction models. The model is used to predict the residential land price, average wage, per capita crime rate, proportion of residential land zoned for lots over 25,000 sq.ft, pupil-teacher ratio by town and so on. Then classical linear regression model is used to predict the housing price.

Introduction

Linear Regression is a method to model the relationship between a set of independent variables X(also knowns as explanatory variables, features, predictors) and a dependent variable Y . This method assumes the relationship between each predictor is linearly related to the dependent variable Y.

$$Y = \beta_0 + \beta_1 X + \epsilon$$

where ϵ is considered as an unobservable random variable that adds noise to the linear relationship and β_0 is the intercept of the linear model. This is the simplest form of linear regression (one variable), we'll call this the simple model.

How do you estimate the coefficients?

- There are many ways to fit a linear regression model
- The method called **least squares** is one of the most common methods

Methodology

The goal is to estimate the coefficients (e.g. β_0 and β_1). We represent the estimates of the coefficients with a "hat" on top of the letter.

$$\hat{\beta}_0, \hat{\beta}_1$$

Once you estimate the coefficients $\hat{\beta}_0, \hat{\beta}_1$ you can use these to predict new values of Y.

Least squares is a method that can estimate the coefficients of a linear model by minimizing the difference between the following:

$$S = \sum_{i=1}^{N} r_i = \sum_{i=1}^{N} (y_i - (\beta_0 + \beta_1 x_i))^2$$ where N is the number of observations.

The solution can be written in compact matrix notation as :

$$\hat{\beta} = (X^T X)^{-1} X^T Y$$

We wanted to show you this in case you remember linear algebra, in order for this solution to exist we need X to be invertible. Of course this requires a few extra assumptions, $X^T X$ must be full rank so that $X^T X$ is invertible, etc. This is important for us because this means that having redundant features in our regression models will lead to poorly fitting (and unstable) models.

Dataset

The Boston Housing data set contains information about the housing values in suburbs of Boston. This dataset was originally taken from the StatLib library which is maintained at Carnegie Mellon University and is now available on the UCI Machine Learning Repository.

Other important things to think about when fitting a linear regression model

---

- **Linearity**. The dependent variable YY is a linear combination of the regression coefficients and the independent variables XX.
- **Constant standard deviation**. The SD of the dependent variable YY should be constant for different values of X.
    - e.g. PTRATIO
- **Normal distribution for errors**. The ϵϵ term we discussed at the beginning are assumed to be normally distributed.
    - $\epsilon_i \sim N(0,\sigma 2)$ϵi~N(0,σ2)
- Sometimes the distributions of responses YY may not be normally distributed at any given value of XX. e.g. skewed positively or negatively.
- **Independent errors**. The observations are assumed to be obtained independently.
    - e.g. Observations across time may be correlated

Training and Test Data sets

Purpose of splitting data into Training/testing sets

Let's stick to the linear regression example:

- We built our model with the requirement that the model fit the data well.
- As a side-effect, the model will fit THIS dataset well. What about new data?
    - We wanted the model for predictions, right?
- One simple solution, leave out some data (for testing) and train the model on the rest
- This also leads directly to the idea of cross-validation, next section.

K-fold Cross-validation as an extension of this idea

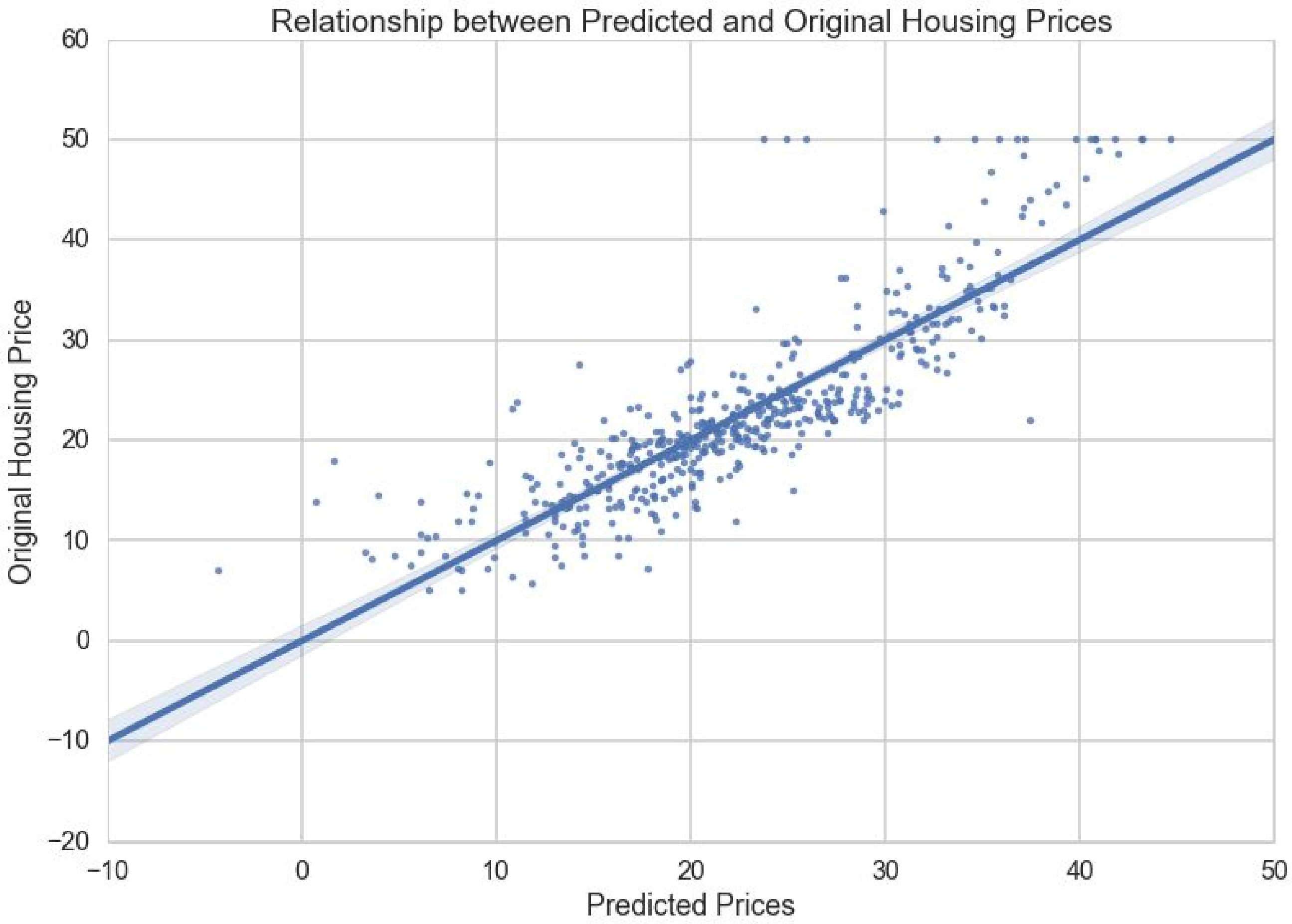
A simple extension of the Test/train split is called K-fold cross-validation.

Here's the procedure:

- randomly assign your n samples to one of K groups. They'll each have about n/k samples
- For each group k:
    - Fit the model (e.g. run regression) on all data excluding the k^{th} group
    - Use the model to predict the outcomes in group k
    - Calculate your prediction error for each observation in k^{th} group

Output

The fitted regression line is shown as below.



Conclusion

The K-Fold cross-validation yields an average prediction error that is smaller than the simple train-test split used previously. Multiple rounds of cross-validation performed on different partitions help limit the problem of overfitting a particular training subset and thus reduce variability of the model.