ML 4: Least Squares Linear Regression

Iustina Ivanova, *Msc student Artificial Intelligence,* email: *ii1n17@soton.ac.uk*

I. Introduction

An example of implementation of linear regression algorithm is described. .

A. Linear Regression

The dataset "Boston Housing" is used to implement algorithm of linear regression. There are 13 values, the price on house depends on all of them.

Linear regression fits a data model that is linear in the model coefficients. The most common type of linear regression is a least-squares fit, which can fit both lines and polynomials, among other models.

The simplest linear model for regression is one that involves a linear combination of the input variables

$$f(\overrightarrow{x}, \overrightarrow{w}) = w_0 + w_1 x_1 + ... + w_N x_N$$

where $x = (x_1, ..., x_N)^T$. This is often known as linear regression. The key property of this model is that it is a linear function of the parameters $w_0, ..., w_N$.

The class of models is considered to be linear combinations of fixed nonlinear functions of the input variables:

$$f(\overrightarrow{x}\overrightarrow{w}) = w_0 + \overrightarrow{w}^t \overrightarrow{x}$$

It is also, however, a linear function of the input variables x_i, and this imposes significant limitations on the model. We therefor extend the class of models by considering linear combinations of fixed nonlinear functions of the input variables, of the form:

$\{\overrightarrow{y}_n, f_n\}_{n=1}^N$, where $\overrightarrow{y} = (\overrightarrow{x}, 1)^T$ and $\overrightarrow{a} = (\overrightarrow{w}, w_0)^T$.

Then model is $f = \overrightarrow{y}^t \overrightarrow{a}$.

We fit function to data set by minimizing a sum-of-squares error function. This error function could be motivated as the maximum likelihood solution under an assumed noise model. The sum-of-squares error function is defined by

$$E = \sum_{n=1}^N \{\overrightarrow{y}_n^t \overrightarrow{a} - f_n\}^2$$
$$E = \sum_{n=1}^N \left\{\left(\sum_{j=1}^{(p+1)} a_j y_{nj}\right) - f_n\right\}^2$$

To find the best $\overrightarrow{a}$ we minimize E - differentiate with respect to each of the unknowns in $\overrightarrow{a}$ and set to zero.

$$\frac{\partial E}{\partial a_i} = 2\sum_{n=1}^N \left\{\sum_{j=1}^{(p+1)} a_j y_{nj} - f_n\right\}(y_{ni})$$

There are (p+1) derivates (with respect to each a_i). Equating them to zero gives (p+1) equations in (p+1) unknowns. We have to solve (p+1) simultaneous equations:

i^{th} row, j^{th} column shown

$$\begin{pmatrix} \cdots & \cdots & \cdots \\ \cdots & \sum_{n=1}^N y_{ni} y_{nj} & \cdots \\ \cdots & \cdots & \cdots \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \cdots \\ a_{p+1} \end{pmatrix} = \begin{pmatrix} \cdots \\ \sum_{n=1}^N f_n y_{ni} \\ \cdots \end{pmatrix}$$

The gradient of log likelihood function takes the form $\Delta_a E = 2\overrightarrow{Y}^t \left(\overrightarrow{Y}\overrightarrow{a} - \overrightarrow{f}\right)$ Equating the gradient to zero gives

$$\overrightarrow{Y}^t \overrightarrow{Y}\overrightarrow{a} = \overrightarrow{Y}^t \overrightarrow{f}$$
$$\overrightarrow{a} = \left(\overrightarrow{Y}^t \overrightarrow{Y}\right)^{-1} \overrightarrow{Y}^t \overrightarrow{f}$$

The implementation of this formula into Matlab Code plot the graphic like this:

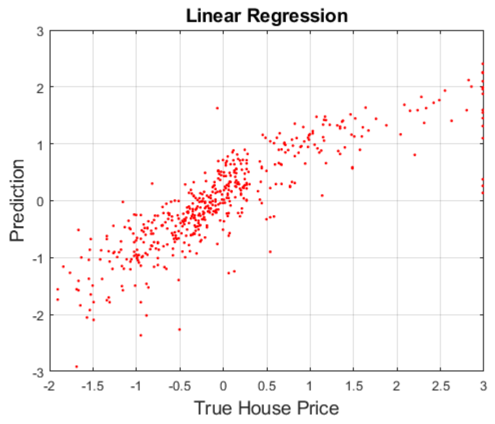

Image 1. The linear regression of Boston Housing data

The data of Boston Housing is splited then to the training set (450 first values) and test set (56 values). Estimation the regression model on the training set and the training and test errors differ:

training error = 0.2637

test error = 0.4480

Implementation of 10-fold cross validation on the data is made to see an average prediction error and an uncertainty on it:

average error: 0.2854

uncertainty: 0.1119

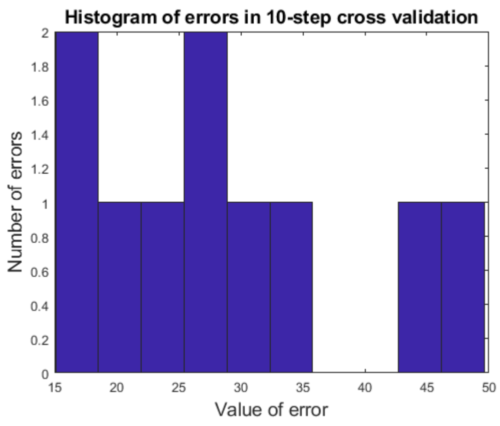

Image 2. Histogram of errors in 10-step cross validation

II. Regression using the CVX Tool:

The least squares regression can be implemented in the cvx tool. Two methods produce nearly same results:

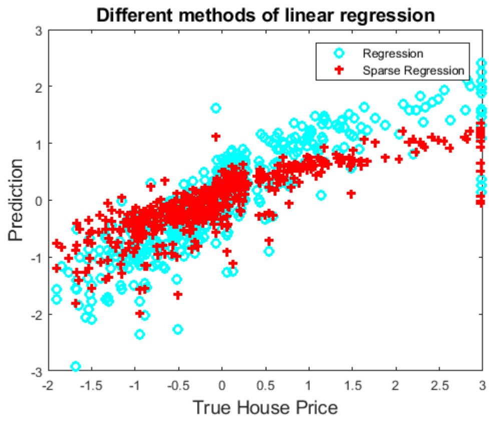

Image 3. The linear squares regression of 2 different methods.

The classical linear regression problem

$$\left\| \overrightarrow{Y} \overrightarrow{a} - \overrightarrow{f} \right\| + \gamma \left\| \overrightarrow{a} \right\|^2$$

The solution to this problem is well-defined if number of training examples $(n) > (p)$ number of features. But in some applications $n << p$, and many of the extracted features $\gamma_1, ..., \gamma_p$ could be irrelevant. We therefore wish to find a model x with many zero coefficients, as illustrated in Image 4.

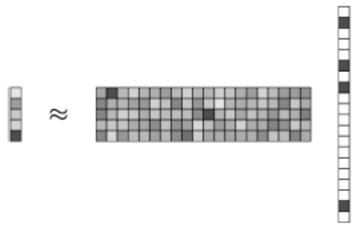

Image 4. The linear regression model $\overrightarrow{y} \approx \overrightarrow{A} \overrightarrow{x}$

The only known algorithm is essentially brute-force evaluation of all possible subsets; the solution cannot be computed in practise for number of features $>= 40$.

A vector $A = (A_1, A_2, ..., A_n$ has γ-norm $\left\| \overrightarrow{A} \right\|_\gamma = \left(\sum_{i=1}^{n} \left| \overrightarrow{A}_i \right|^\gamma \right)^{\frac{1}{\gamma}}$ Taking $\gamma = 8$ we can calculate that relevant variables are 6,11,13, that means,that this parameters are the most important for price.

6. RM average number of rooms per dwelling
11. PTRATIO pupil-teacher ratio by town
13. LSTAT lower status of the population

Calculation of different values of γ shows that solution cannot be computed, when $\gamma = 17$

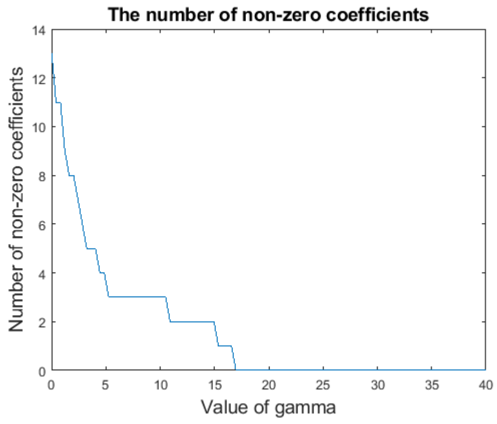

Image 5. Dependence of number of coefficients of value of gamma.

III. Conclusion

In this work the linear least squares regression was implemented. The most valuable parameters for cost on houses are number of rooms, good studying facilities and population. The result can be considered as good.

References

[1] Christopher M. Bishop , *Pattern Recognition and Machine Learning,2006* .

[2] Richard O. Duda, Peter E. Hart, David G. Stork, *Pattern Classification, second edition, 2006.* .