

LAB ACTIVITY 5:

Optimized Linear Regression

CARANDO, Paolo Creas
GOMENDOZA, Frank Gerald
LIM, Theo Paolo
LORENA, Neil John
ZABAT, Juan Paolo

Abstract— In this laboratory experiment, the group will learn about supervised regression. Basically supervised learning is the task of inferring a function from labelled training data. The function will be used is minFunc. The difference between the previous regression, a supervised learning will not be limited to the values true for the linear equation and also the accuracy will improve.

Keywords—Linear regression, optimization, supervised learning

I. INTRODUCTION

On the previous laboratory experiment, linear regression is already being taught. Usually linear regression is used to predict a certain data. For example, in predicting the house price depending on the number of rooms and area. The linear regression is implemented to develop a line that will represent the original data. This line will be implemented to predict values, with respect to a certain input. Theta 0 and Theta 1 are the value that represent the y-intercept and slope of the line. $J(\theta)$ will be minimized.

The cost function $J(\theta)$ is not always the hypothesis, as well as resulting prediction, is precise. Based on our previous experiment, there are 3 kinds of fit. Under fit, just right and overfit. Similarly, underfit and overfit can still be present. Thus $J(\theta)$ is not ideal to improve the accuracy of prediction.

A supervised learning algorithm analyzes the training data, and produces a function which can map the fit of the data.

II. OBJECTIVES

The experiment aims to achieve the following objectives

- To be able to produce a plot showing the plot of the linear regression
- To be able to produce a minFunc code

III. DATA AND RESULTS

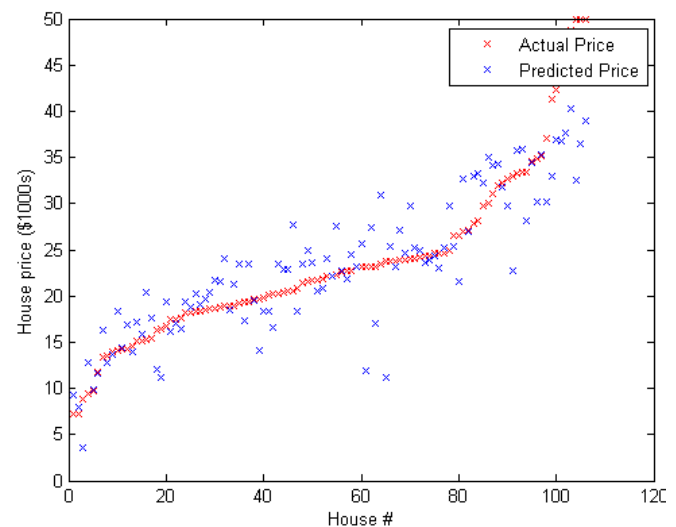


Figure 1. Predicted price using optimized linear regression

IV. ANALYSIS AND CONCLUSION

The linear regression using minFunc to optimize the fit of the prediction. $J(\theta)$ is also used as well as the deviation to attempt to minimize it while minimizing the cost function. This will result to a prediction that will be closer to the actual values.

the linear regression is optimized using the minFunc package. Instead of only relying on the $J(\theta)$, we also take a look at the deviation and attempts to minimize it while also minimizing $J(\theta)$. In this way, predicted values will be closer to the actual values. It will first determine the type of training data. It will then gather the data set and will determine the input representation of the learned function. The prediction will depend on the learned function and it will improve if it will learn more training data.

APPENDIX

Code

```
% Run using ex1a_linreg.m
function [f,g] = linear_regression(theta, X,y)
%
% Arguments:
%   theta - A vector containing the parameter values to optimize.
%   X - The examples stored in a matrix.
%       X(i,j) is the i'th coordinate of the j'th example.
%   y - The target value for each example. y(j) is the target for example j.
%
m=size(X,2);
n=size(X,1);

f=0;
g=zeros(size(theta));

%
% TODO: Compute the linear regression objective by looping over the examples in X.
%       Store the objective function value in 'f'.
%
% TODO: Compute the gradient of the objective with respect to theta by looping over
%       the examples in X and adding up the gradient for each example. Store the
%       computed gradient in 'g'.

%%% YOUR CODE HERE %%%

f= .5 * (y-theta * X')*(y-theta * X')';
g= X * X' *theta-X * y';

end
```