Boston Housing Exercise 8 and 9

Table of Contents

Setup

```
%load data and explore variables
clear all; close all;
data = importdata('boston.mat');
[dataS dataN] = size(data);
% other setup;
count = 20; % iteration counter
rng(8); % fix time for random data shuffle (same value produced when
 re-run)
% data split (1/3 test and 2/3 train)
trainSize = round(dataS*2/3);
testSize = dataS-trainSize;

% Boston data description of variables:
%-last column is our unknown price of proprty to b learned
% Variables; (1st 13 var are regressors and the 14th var is our
 dependant
% variable to be learned)
% There are 14 attributes in each case of the dataset. They are:
% CRIM - per capita crime rate by town
% ZN - proportion of residential land zoned for lots over 25,000
 sq.ft.
% INDUS - proportion of non-retail business acres per town.
% CHAS - Charles River dummy variable (1 if tract bounds river; 0
 otherwise)
% NOX - nitric oxides concentration (parts per 10 million)
% RM - average number of rooms per dwelling
% AGE - proportion of owner-occupied units built prior to 1940
% DIS - weighted distances to five Boston employment centres
% RAD - index of accessibility to radial highways
% TAX - full-value property-tax rate per $10,000
% PTRATIO - pupil-teacher ratio by town
% B - 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
% LSTAT - % lower status of the population
% MEDV - Median value of owner-occupied homes in $1000's
```

Part 1 (a) - Naive Regression- Predicting with the mean y-value

```
One = ones(dataS, 1); % bias vector
w1 = zeros(count,1);  % initialise weight vector
MseTrain1 = zeros(count,1);
MseTest1 = zeros(count,1);

for i = 1:count

    % Shuffle last column of data and split into test and training
 sets

    dataShuffle = data(randperm(dataS), 14);

    dataTrain = dataShuffle(1:trainSize);
    dataTest = dataShuffle(trainSize + 1:dataS);

    % perform linear regression using only the bias term
    w1(i,1) = One(1:trainSize)\dataTrain(:);

    % calculate mean squared error on training and test sets
    MseTrain1(i,1) = sum((dataTrain-(One(1:trainSize)*w1(i,1))).^2)/
size(dataTrain,1);
    MseTest1(i,1) = sum((dataTest-(One(trainSize
+1:dataS)*w1(i,1))).^2)/size(dataTest,1);

end

% calculate the mean errors over all the randomly generated datasets
meanMSEtrain = mean(MseTrain1);
meanMSEtest = mean(MseTest1);
sdMSEtrain = std(MseTrain1);
sdMSEtest = std(MseTest1);
```

Part 2 - LR- single attribute and a bias term

```
dim = 13; %number of regressors
% initailise weight and MSE arrays:
w2 = zeros(2,dim);
MseTrain2 = zeros(count,dim);
MseTest2 = zeros(count,dim);

for j = 1:dim % iterate through all 13 variables
     for i = 1:count % repat for 20 random draws, similar to prev
 parts

        % Shuffle all data and split into separate test and training
 sets
```

```
        dataShuffle = data(randperm(dataS), :);% overwrite var from
 Part 2

        dataTrain = dataShuffle(1:trainSize,:);% overwrite vars from
 Part 2
        dataTest = dataShuffle(trainSize + 1:dataS,:);

        % perform linear regression with the bias term and and one
 regressor
        w2(:,j) = [One(1:trainSize,1),
 dataTrain(:,j)]\dataTrain(:,14);

        % calculate mean squared error on training and test sets
        MseTrain2(i,j) = sum((dataTrain(1:trainSize,14)-
([One(1:trainSize), dataTrain(:,j)]*w2(:,j))).^2)/trainSize;
        MseTest2(i,j) = sum((dataTest(:,14)-([One(trainSize+1:dataS),
 dataTest(:,j) ]*w2(:,j))).^2)/testSize;

    end
end

meanMSEtrain2 = mean(MseTrain2);
meanMSEtest2 = mean(MseTest2);
sdMSEtrain2 = std(MseTrain2);
sdMSEtest2 = std(MseTest2);

% mean taken column-wise so output of the above is a 1 by 13 vector of
 mean
% MSE over 20 runs for LR with the 13 diff variables. These MSE values
% differ depending on the variables used, with min MSE reached for var
 13
% (approx. 39) and followed by var 6 (appox 43).
% Indicating that if we wanted to use LR model on the Boston dataset
 we
% should focus on var 13 first, this should then be followed by var 6
 and
% so on in ascending order of the MSE values. Max MSE is reached for
 var 4,
% thus this should be used last.

minMSEtrain2 = min(meanMSEtrain2);
minMSEtest2 = min(meanMSEtest2);
maxMSEtrain2 = max(meanMSEtrain2);
maxMSEtest2 = max(meanMSEtest2);
meanAllMSEtrain2 = mean(meanMSEtrain2);
meanAllMSEtest2 = mean(meanMSEtest2);
```

Part 3 - LR using all attributes simultaneously

```
% initailise weight and MSE arrays:
w3 = zeros(dataN,count); % size 14 by 20
MseTrain3 = zeros(count,1);
MseTest3 = zeros(count,1);
```

```
    for i = 1:count % repat for 20 random draws, similar to part 1

        % Shuffle all data and split into separate test and training
 sets

        dataShuffle = data(randperm(dataS), :);% overwrite var from
 Part 1

        dataTrain = dataShuffle(1:trainSize,:);  % overwrite vars from
 Part 1
        dataTest = dataShuffle(trainSize + 1:dataS,:);

        % perform linear regression with the bias term and and one
 regressor
        w3(:,i) = [One(1:trainSize,1),
 dataTrain(:,1:13)]\dataTrain(:,14);

        % calculate mean squared error on training and test sets
        MseTrain3(i,1) = sum((dataTrain(1:trainSize,14)-
([One(1:trainSize), dataTrain(:,1:13)]*w3(:,i))).^2)/trainSize;
        MseTest3(i,1) = sum((dataTest(:,14)-([One(trainSize+1:dataS),
 dataTest(:,1:13) ]*w3(:,i))).^2)/testSize;

    end

% calculate summary results over 20 runs:
meanMSEtrain3 = mean(MseTrain3);
meanMSEtest3 = mean(MseTest3);
sdMSEtrain3 = std(MseTrain3);
sdMSEtest3 = std(MseTest3);

% significant reduction in MSE can be observed. The mean MSE over the
 20
% runs reduced to approx. 23 (test) and a standard deviation of under
 4.

% save required results
save('ex9VARS','meanMSEtrain3','meanMSEtest3','sdMSEtrain3','sdMSEtest3','meanMSEt
```

OUTPUT summary table of results for Parts 1,2,3

```
table([meanMSEtrain; minMSEtrain2; maxMSEtrain2; meanAllMSEtrain2;
 meanMSEtrain3],...
     [meanMSEtest; minMSEtest2; maxMSEtest2; meanAllMSEtest2;
 meanMSEtest3], ...
     'VariableNames', {'Train' 'Test'},...
     'RowNames', {'Part 1 - Mean value',...
     'Part 2 - LR, 1 variable - min MSE', ...
     'Part 2 - LR, 1 variable - max MSE', ...
     'Part 2 - LR, 1 variable - mean MSE','Part 3 - All variables'})
```

```
ans =

                                       Train      Test
                                       _____    _____

    Part 1 - Mean value                85.482   82.461
    Part 2 - LR, 1 variable - min MSE  38.115   39.606
    Part 2 - LR, 1 variable - max MSE  81.635   82.897
    Part 2 - LR, 1 variable - mean MSE 66.252   68.752
    Part 3 - All variables             22.068   22.939
```

Published with MATLAB® R2016b