Ict for Smart Societies

Politecnico di Torino

ICT FOR HEALTH LABORATORY

Authors:

Putina Andrian s226673

Professor:

Visentin Monica

November 20, 2016

Contents

1 Lab1: Linear Regression & Parkinson

The aim of the laboratory is understanding the basic idea of the regression and apply different techniques to obtain the desired results. In particular, it will be considered a database containing a real Parkinson's Disease dataset and different regression solutions will by applied:

- MSE (Minimum Square Error)
- Gradient Algorithm
- Steepest Descent Algorithm

Source: https://archive.ics.uci.edu/ml/machine-learning-databases/parkinsons/telemonitoring/

1.1 Prepare the Data

The first script takes as input variable the total number of pacients `Npacients` and converts the fourth column (test_time) into its integer part. Furthermore creates a new matrix containing, for all the dates, only the mean of the measurements for each pacient so at the end the new matrix has one row for each day (with the mean of the measured values of the features) for each pacient.

```
close all
clear all
%Lab 1 - ICT HEALTH - Prepare the data

load('updrs.mat');

Npacients = 42;
parkinsonsupdrs(:,4) = abs(floor(parkinsonsupdrs(:,4)));

finalMatrix = [];
for pacient = 1: Npacients

    rowPacient = parkinsonsupdrs(:, 1) == pacient;
    matrixPacient = parkinsonsupdrs(rowPacient,:);

    count = 0;
    for day = 1:max(parkinsonsupdrs(:,4))+1
       indx = matrixPacient(:,4) == day-1;
       if (any(indx(:) == 1))
           count = count+1;
           meanMatrixPacient(count,:) = mean(matrixPacient(indx,:));
       end
    end

    finalMatrix = [finalMatrix; meanMatrixPacient];
    meanMatrixPacient=[];

end
save('finalMatrix','finalMatrix');
```

The next step is to divide the pacients into two groups. The first sub-group will be used for the training phase of the algorithm while the second one will be used for the testing part. Anyway the data has to have zero mean and variance equal to 1 (EXPLAIN WHY!!!). The easiest way is to subscract the mean of each feature to all the pacients and divide by the standard deviation. Obviously the mean and the standard deviation is computed on training data we decided before. The data is now normalized so has zero mean and variance = 1

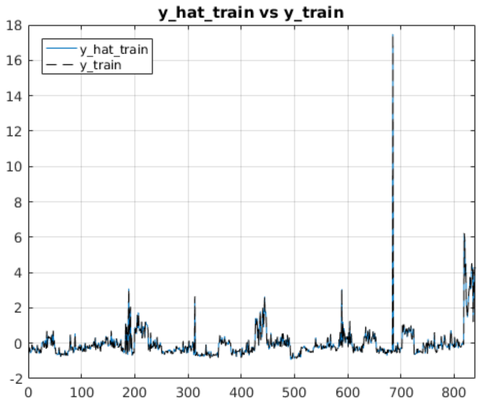
```
close all
clear all
%Lab 1 - ICT HEALTH - Prepare data for train

load('finalMatrix.mat');

Npacients = 36;
rowPacients_train = finalMatrix(:,1) <= Npacients;
rowPacients_test =  finalMatrix(:,1) > Npacients;

data_train = finalMatrix(rowPacients_train,:);
    m_data_train = mean(data_train,1);
    v_data_train = var(data_train,1);
    std_data_train = std(data_train,1);

    data_train_norm = data_train(:,5:22) - m_data_train(:,5:22);
    data_train_norm = data_train_norm ./ std_data_train(:,5:22);
    data_train_norm = [data_train(:,1:4), data_train_norm];

data_test = finalMatrix(rowPacients_test,:);
    data_test_norm = data_test(:,5:22) - m_data_train(:,5:22);
    data_test_norm = data_test_norm ./ std_data_train(:,5:22);
    data_test_norm = [data_test(:,1:4), data_test_norm];

save('data_train_norm','data_train_norm');
save('data_test_norm','data_test_norm');
```

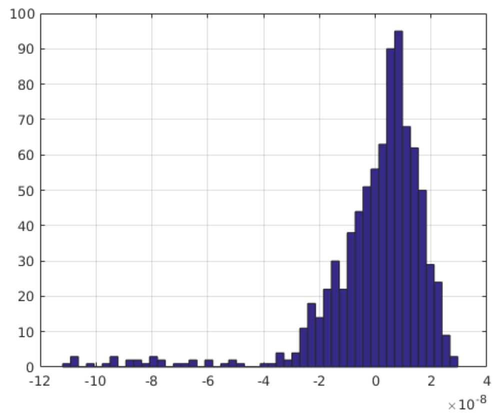
1.2 Perform Regression with MSE

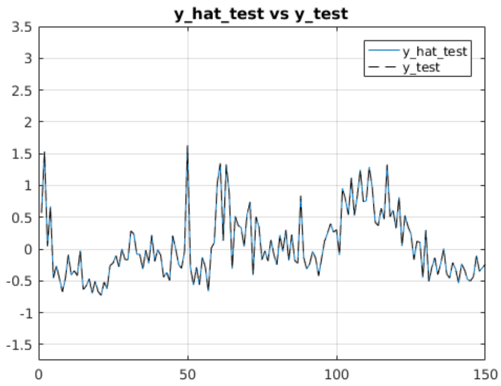
QUALCOSA DI SCRITTO SULLA TEORIA

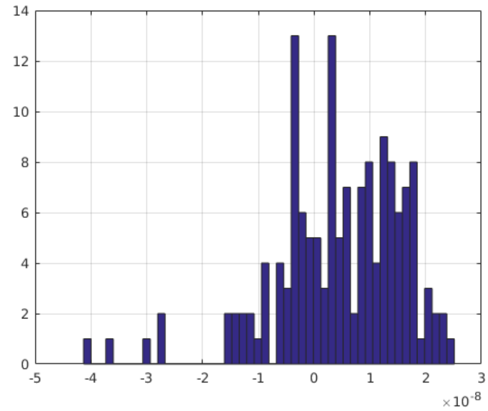
```
close all
clear all
%Lab 1 - ICT HEALTH - Perform regression - MSE

load('data_train_norm.mat');
load('data_test_norm.mat');

F0 = 7;
flag_features = 4; %first features

y_train = data_train_norm(:,F0);
y_test = data_test_norm(:,F0);

X_train = data_train_norm(:,flag_features:end);
X_test = data_test_norm(:,flag_features:end);
```

```
X_train(:,F0-flag_features) = [];
X_test(:,F0-flag_features)=[];

% Estimate a_hat
a_hat = inv(transpose(X_train)*X_train)*transpose(X_train)*y_train;
y_hat_train = X_train * a_hat;
y_hat_test = X_test * a_hat;

figure
plot(y_hat_train)
hold on
plot(y_train, '--k')
axis([0 840 -2 18])
grid on
legend('y\_hat\_train','y\_train', 'Location', 'northwest')
title('y\_hat\_train vs y\_train')

figure
plot(y_test)
hold on
plot(y_hat_test, '--k')
axis([0 150 -1.75 3.5])
grid on
legend('y\_hat\_test','y\_test')
title('y\_hat\_test vs y\_test')

figure
hist(y_hat_train - y_train, 50)
grid on

figure
hist(y_hat_test - y_test, 50)
grid on
```






y_hat_test vs y_test
y_hat_test
y_test
3.5
3
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
0
50
100
150


14
12
10
8
6
4
2
0
-5
-4
-3
-2
-1
0
1
2
3
×10^-8