
ICT4Health Lab 1 - Regression

Table of Contents

Data Preparation	1
Regression for feature F0 = 7	2
Regression for feature F0 = 5	7

In the given data for Parkinson's disease patients, we estimate the Unified Parkinson's Disease Rating Scale (UPDRS):
I. the total UPDRS value (column 7), II. the motor UPDRS (column 5), using other features as these two features are expensive to measure

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

Data Preparation

```
close all;
clear all;
clc;

% Data is first imported in matlab and saved as numeric matrix
load('updrs.mat'); % loading numeric matrix stored saved before
updrs = sortrows(parkinsonsupdrs, [1,4]); % sorting the data matrix
using                                     % columns 1 and 4

[nrows, ncols] = size(updrs);
new_row = [];
data=[];
r_count=1;

% loop for aggregating data rows with same time (column 4)
for i=1:nrows-1
    if(ceil(updrs(i,4)) ~= ceil(updrs(i+1,4)))
        new_row = mean(updrs(r_count:i,:));
        data = [data;new_row];
        r_count=i+1;
    end
end
new_row = mean(updrs(r_count:end,:));
data = [data;new_row]; % new matrix with single data for each time
save('parkinsonsdat_for_regression.mat','data');
[nrows, ncols] = size(data);

% Selecting training set data range from given data
for i=1:nrows
    if data(i,1) > 36      % Data until patient number 36 is selected
        break
    end
    row_cut = i;
end
data_train = data(1:row_cut, :);
```

```
data_test = data(row_cut+1:end,:);

% Standardising training and testing set data
m_data_train = mean(data_train,1);
v_data_train = std(data_train,1);
data_train_norm = (data_train - m_data_train)./v_data_train;

m_data_test = mean(data_test,1);
v_data_test = std(data_test,1);
data_test_norm = (data_test - m_data_test)./v_data_test;

% Testing and training data are standardised after splitting
% to avoid the inclusion of future information while developing model
```

Regression for feature F0 = 7

```
F0 = 7;
Y_train = data_train_norm(:,F0);
X_train = data_train_norm;
X_train(:,F0) = [];

Y_test = data_test_norm(:,F0);
X_test = data_test_norm;
X_test(:,F0) = [];

MSE Algorithm

a_mse = inv(transpose(X_train)*X_train)*transpose(X_train)*Y_train;

yhat_train = X_train*a_mse;
yhat_test = X_test*a_mse;

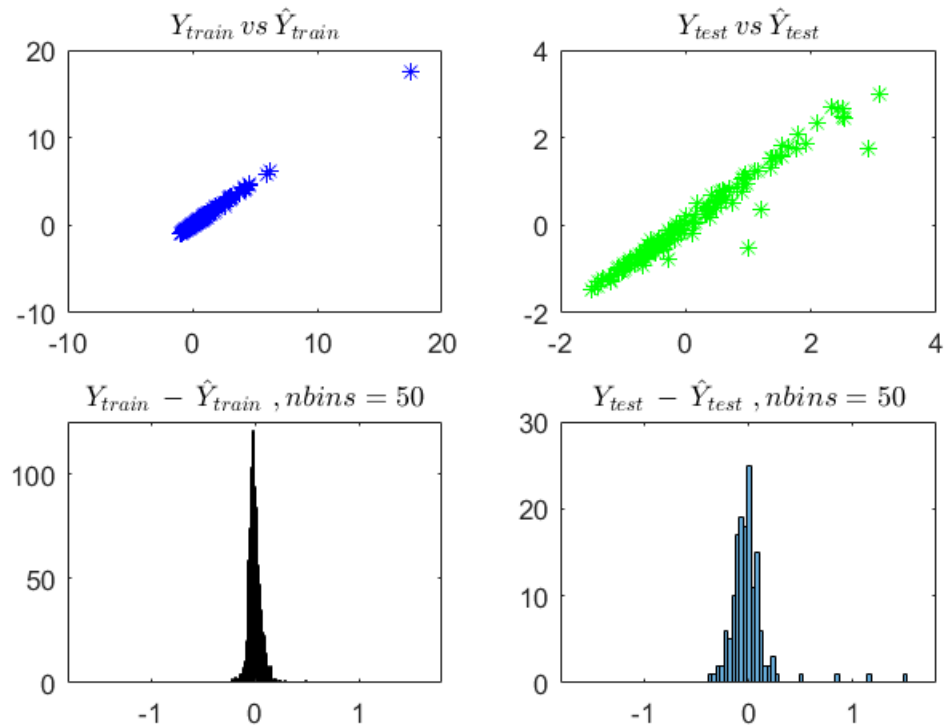
figure
subplot(2,2,1)
plot(Y_train,yhat_train,'b*');
title('$Y_{train} \setminus, vs \setminus, \hat{Y}_{train}$','Interpreter','latex')

subplot(2,2,2)
plot(Y_test,yhat_test,'g*');
title('$Y_{test} \setminus, vs \setminus, \hat{Y}_{test}$','Interpreter','latex')

nbins = 50;
subplot(2,2,3)
histogram(Y_train-yhat_train,nbins);
title('$Y_{train} \setminus, - \setminus, \hat{Y}_{train} \setminus, \setminus, ,nbins=50$','Interpreter','latex')
axis([-1.8 1.8 0 125])

subplot(2,2,4)
histogram(Y_test-yhat_test,nbins);
title('$Y_{test} \setminus, - \setminus, \hat{Y}_{test} \setminus, \setminus, ,nbins=50$','Interpreter','latex')
axis([-1.8 1.8 0 30])
suptitle('MSE Algorithm [F0 = 7]')
```

MSE Algorithm [F0 = 7]



Gradient Algorithm

```

rng('default');
lambda = 1.07e-4;
epsi = 0.5e-15;

num_its = 100000;

% Large variables and for-loop is used for monitoring and debugging
a_grad = zeros(length(a_mse),num_its);
a_grad(:,1) = randn(length(a_mse),1);
a_grad(:,2) = ones(length(a_mse),1);
grad = zeros(length(a_mse),num_its);
flag=1;
for i=1:num_its-1
    grad(:,i) = ((-2*transpose(X_train)*Y_train) +
    (2*transpose(X_train)*X_train*a_grad(:,i)));
    a_grad(:,i+1) = a_grad(:,i) - (lambda*grad(:,i));

    if (((a_grad(:,i+1) - a_grad(:,i)))'*(a_grad(:,i+1) - a_grad(:,i)))
    < epsi)
        it_fin = i;
        break
    end
    it_fin = i;
end
end

```

```

yhat_train_gd = X_train*a_grad(:,it_fin);
yhat_test_gd = X_test*a_grad(:,it_fin);

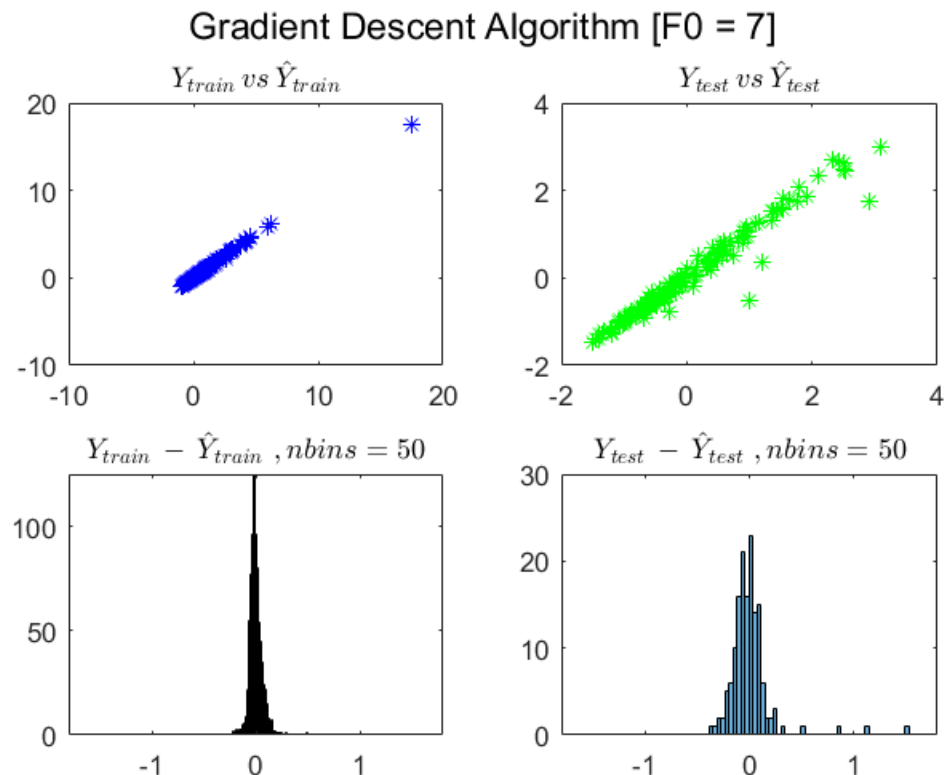
figure
subplot(2,2,1)
plot(Y_train,yhat_train_gd,'b*');
title('$Y_{train} \setminus, vs \setminus, \hat{Y}_{train}$','Interpreter','latex')

subplot(2,2,2)
plot(Y_test,yhat_test_gd,'g*');
title('$Y_{test} \setminus, vs \setminus, \hat{Y}_{test}$','Interpreter','latex')

nbins = 50;
subplot(2,2,3)
histogram(Y_train-yhat_train_gd,nbins);
title('$Y_{train} \setminus, - \setminus, \hat{Y}_{train} \setminus, \setminus, ,nbins=50$','Interpreter','latex')
axis([-1.8 1.8 0 125])

subplot(2,2,4)
histogram(Y_test-yhat_test_gd,nbins);
title('$Y_{test} \setminus, - \setminus, \hat{Y}_{test} \setminus, \setminus, ,nbins=50$','Interpreter','latex')
axis([-1.8 1.8 0 30])
suptitle('Gradient Descent Algorithm [F0 = 7]')

```



Steepest Descent Algorithm

```

epsi_sd = 1e-14;
num_its_sd = 100000;

a_grad_sd = zeros(length(a_mse),num_its_sd);
a_grad_sd(:,1) = randn(length(a_mse),1);
a_grad_sd(:,2) = ones(length(a_mse),1);
grad_sd = zeros(length(a_mse),num_its_sd);

for i=1:num_its_sd-1
    grad_sd(:,i) = -2*X_train'*Y_train +
        2*X_train'*X_train*a_grad_sd(:,i);
    hess = 4*X_train'*X_train;

    denom = grad_sd(:,i)'*hess*grad_sd(:,i);
    num = grad_sd(:,i)'*grad_sd(:,i);
    a_grad_sd(:,i+1) = a_grad_sd(:,i) - (num/denom)*grad_sd(:,i);

    if (((a_grad_sd(:,i+1) - a_grad_sd(:,i))'*(a_grad_sd(:,i+1) -
        a_grad_sd(:,i))) < epsi_sd)
        it_fin_sd = i;
        break
    end

    it_fin_sd = i;
end

yhat_train_sd = X_train*a_grad_sd(:,it_fin_sd);
yhat_test_sd = X_test*a_grad_sd(:,it_fin_sd);

figure
subplot(2,2,1)
plot(Y_train,yhat_train_sd,'b*');
title('$Y_{train} \setminus, vs \setminus, \hat{Y}_{train}$','Interpreter','latex')

subplot(2,2,2)
plot(Y_test,yhat_test_sd,'g*');
title('$Y_{test} \setminus, vs \setminus, \hat{Y}_{test}$','Interpreter','latex')

nbins = 50;
subplot(2,2,3)
histogram(Y_train-yhat_train_sd,nbins);
title('$Y_{train} \setminus, - \setminus, \hat{Y}_{train} \setminus, \setminus, ,nbins=50$', 'Interpreter','latex')
axis([-1.8 1.8 0 125])

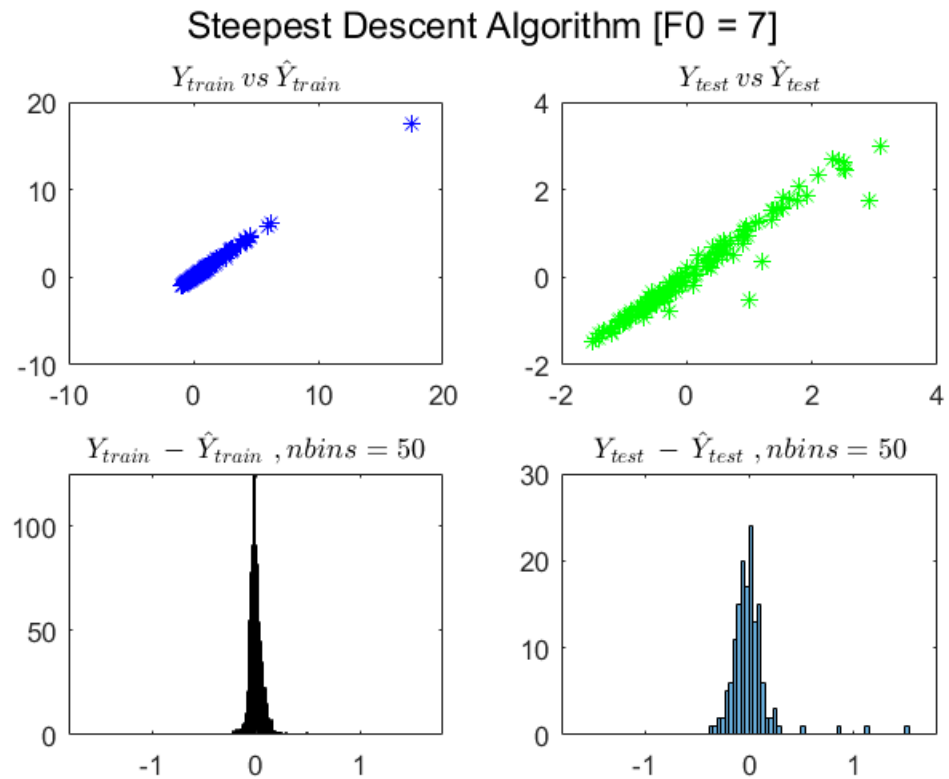
subplot(2,2,4)
histogram(Y_test-yhat_test_sd,nbins);
title('$Y_{test} \setminus, - \setminus, \hat{Y}_{test} \setminus, \setminus, ,nbins=50$', 'Interpreter','latex')
axis([-1.8 1.8 0 30])
suptitle('Steepest Descent Algorithm [F0 = 7]')

```

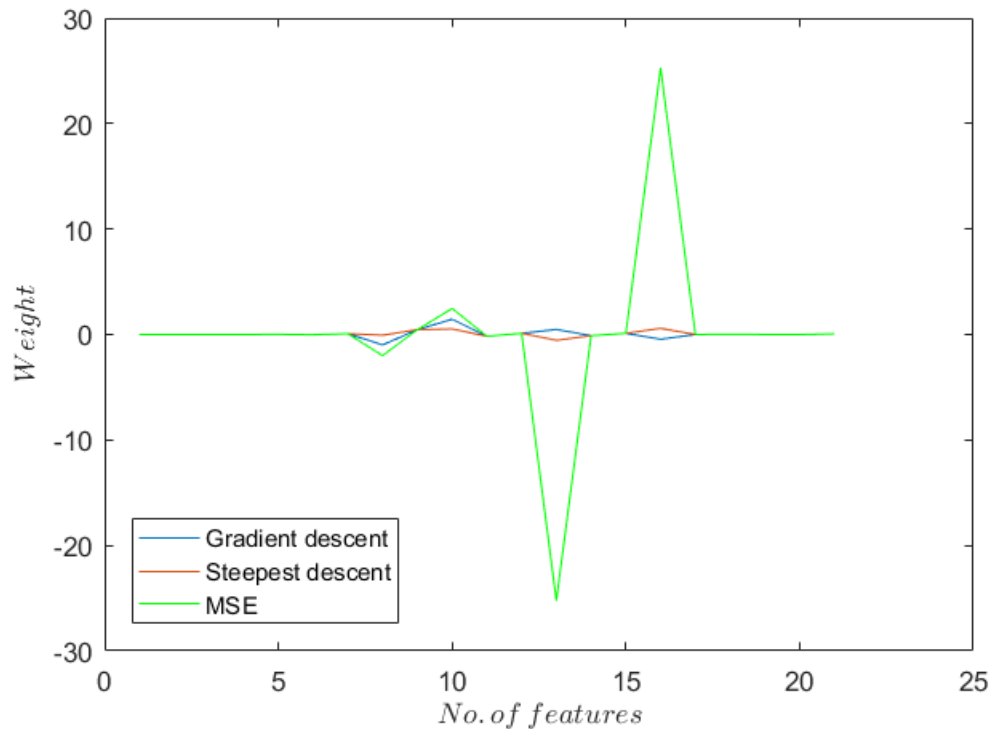
```

figure
xs = 1:length(a_mse);
plot(xs,a_grad(:,it_fin),xs,a_grad_sd(:,it_fin_sd),xs,a_mse,'g')
suptitle('Weights for each feature (a) [F0 = 7]')
xlabel('$No.\,of\,features$', 'Interpreter', 'latex')
ylabel('$Weight$', 'Interpreter', 'latex')
legend('Gradient descent', 'Steepest
descent', 'MSE', 'Location', 'southwest')

```



Weights for each feature (a) [F0 = 7]



Regression for feature F0 = 5

```

F0 = 5;
Y_train = data_train_norm(:,F0);
X_train = data_train_norm;
X_train(:,F0) = [];

Y_test = data_test_norm(:,F0);
X_test = data_test_norm;
X_test(:,F0) = [];

MSE Algorithm

a_mse = inv(transpose(X_train)*X_train)*transpose(X_train)*Y_train;

yhat_train = X_train*a_mse;
yhat_test = X_test*a_mse;

figure
subplot(2,2,1)
plot(Y_train,yhat_train,'b*');
title('$Y_{train} \setminus, vs \setminus, \hat{Y}_{train}$','Interpreter','latex')

subplot(2,2,2)
plot(Y_test,yhat_test,'g*');
title('$Y_{test} \setminus, vs \setminus, \hat{Y}_{test}$','Interpreter','latex')

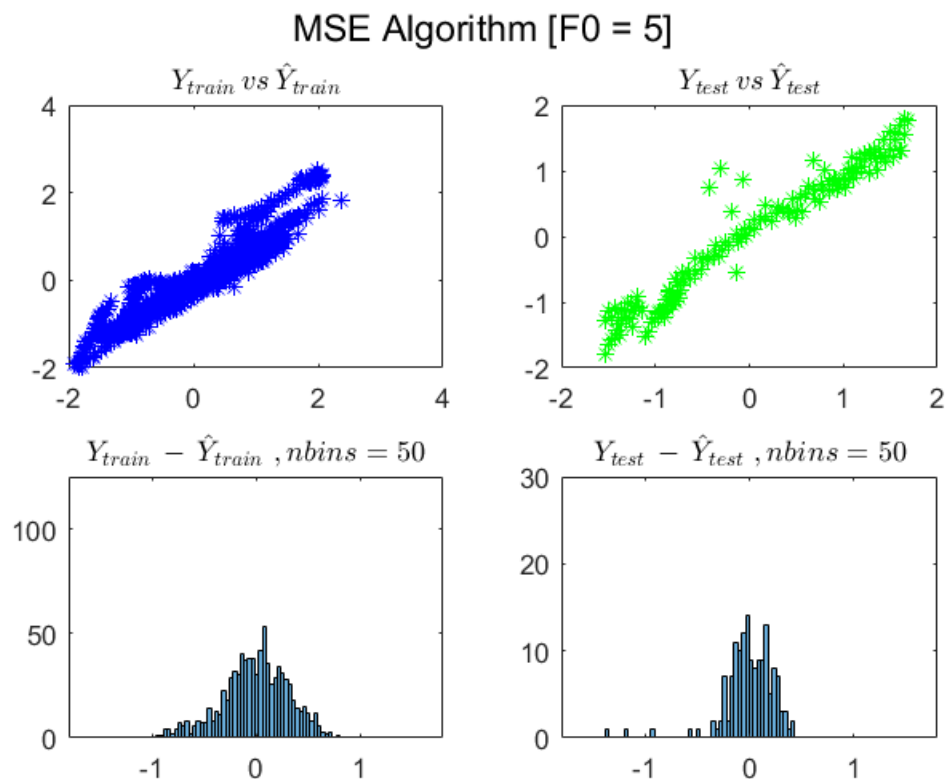
```

```

nbins = 50;
subplot(2,2,3)
histogram(Y_train-yhat_train,nbins);
title('$Y_{train} \setminus, - \setminus, \hat{Y}_{train} \setminus, \setminus, ,nbins=50$', 'Interpreter', 'latex')
axis([-1.8 1.8 0 125])

subplot(2,2,4)
histogram(Y_test-yhat_test,nbins);
title('$Y_{test} \setminus, - \setminus, \hat{Y}_{test} \setminus, \setminus, ,nbins=50$', 'Interpreter', 'latex')
axis([-1.8 1.8 0 30])
suptitle('MSE Algorithm [F0 = 5]')

```



Gradient Algorithm

```

rng('default');
lambda2 = 1.5e-5;
epsi = 1e-12;

num_its = 100000;

% Large variables and for-loop is used for monitoring and debugging
a_grad = zeros(length(a_mse),num_its);
a_grad(:,1) = randn(length(a_mse),1);
a_grad(:,2) = ones(length(a_mse),1);

```



```
grad = zeros(length(a_mse),num_its);
flag=1;
for i=1:num_its-1
    grad(:,i) = ((-2*transpose(X_train)*Y_train) +
    (2*transpose(X_train)*X_train*a_grad(:,i)));
    a_grad(:,i+1) = a_grad(:,i) - (lambda2*grad(:,i));

    if (((a_grad(:,i+1) - a_grad(:,i))'*(a_grad(:,i+1) - a_grad(:,i)))
    < epsi)
        it_fin = i;
        break
    end
    it_fin = i;
end

yhat_train_gd = X_train*a_grad(:,it_fin);
yhat_test_gd = X_test*a_grad(:,it_fin);

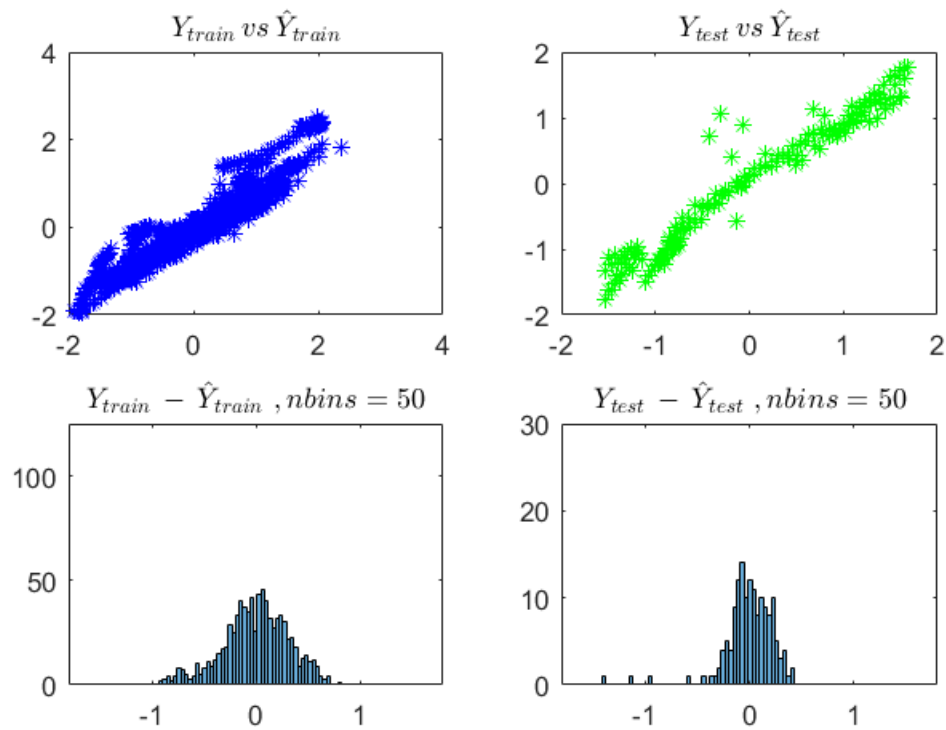
figure
subplot(2,2,1)
plot(Y_train,yhat_train_gd,'b*');
title('$Y_{train} \setminus, \text{vs} \setminus, \hat{Y}_{train}$','Interpreter','latex')

subplot(2,2,2)
plot(Y_test,yhat_test_gd,'g*');
title('$Y_{test} \setminus, \text{vs} \setminus, \hat{Y}_{test}$','Interpreter','latex')

nbins = 50;
subplot(2,2,3)
histogram(Y_train-yhat_train_gd,nbins);
title('$Y_{train} \setminus, - \setminus, \hat{Y}_{train} \setminus, \setminus, ,nbins=50$','Interpreter','latex')
axis([-1.8 1.8 0 125])

subplot(2,2,4)
histogram(Y_test-yhat_test_gd,nbins);
title('$Y_{test} \setminus, - \setminus, \hat{Y}_{test} \setminus, \setminus, ,nbins=50$','Interpreter','latex')
axis([-1.8 1.8 0 30])
suptitle('Gradient Descent Algorithm [F0 = 5]')
```

Gradient Descent Algorithm [F0 = 5]



Steepest Descent Algorithm

```

epsi_sd = 1e-14;
num_its_sd = 100000;

a_grad_sd = zeros(length(a_mse),num_its_sd);
a_grad_sd(:,1) = randn(length(a_mse),1);
a_grad_sd(:,2) = ones(length(a_mse),1);
grad_sd = zeros(length(a_mse),num_its_sd);

for i=1:num_its_sd-1
    grad_sd(:,i) = -2*X_train'*Y_train +
        2*X_train'*X_train*a_grad_sd(:,i);
    hess = 4*X_train'*X_train;

    denom = grad_sd(:,i)'*hess*grad_sd(:,i);
    num = grad_sd(:,i)'*grad_sd(:,i);
    a_grad_sd(:,i+1) = a_grad_sd(:,i) - (num/denom)*grad_sd(:,i);

    if (((a_grad_sd(:,i+1) - a_grad_sd(:,i)))'*(a_grad_sd(:,i+1) -
        a_grad_sd(:,i))) < epsi_sd)
        it_fin_sd = i;
        break
    end

    it_fin_sd = i;

```

```
end

yhat_train_sd = X_train*a_grad_sd(:,it_fin_sd);
yhat_test_sd = X_test*a_grad_sd(:,it_fin_sd);

figure
subplot(2,2,1)
plot(Y_train,yhat_train_sd,'b*');
title('$Y_{train} \setminus, vs \setminus, \hat{Y}_{train}$','Interpreter','latex')

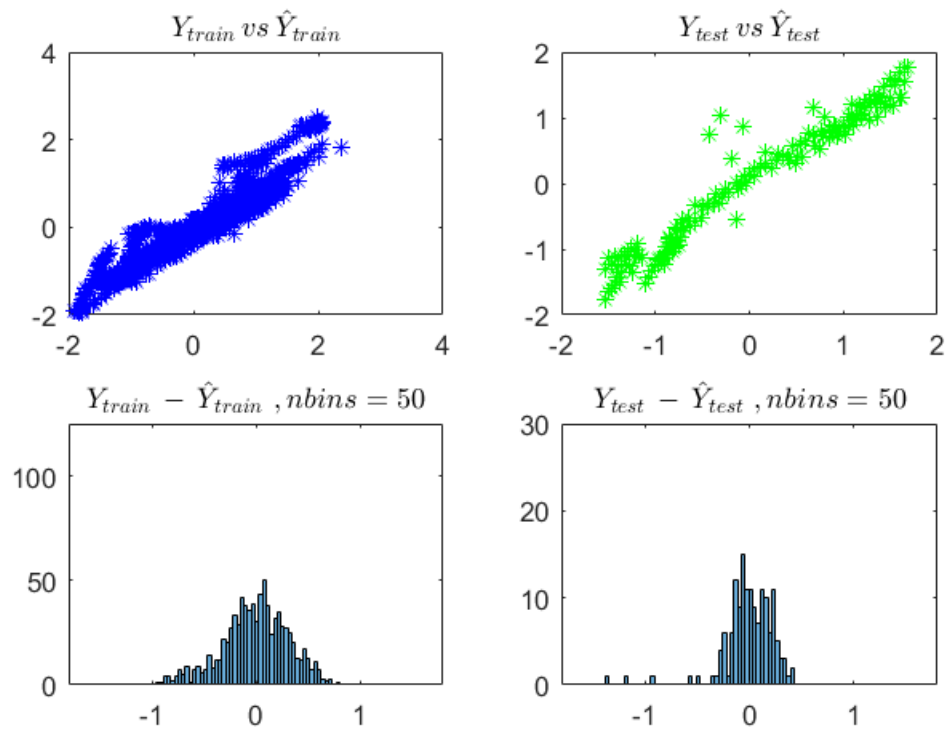
subplot(2,2,2)
plot(Y_test,yhat_test_sd,'g*');
title('$Y_{test} \setminus, vs \setminus, \hat{Y}_{test}$','Interpreter','latex')

nbins = 50;
subplot(2,2,3)
histogram(Y_train-yhat_train_sd,nbins);
title('$Y_{train} \setminus, - \setminus, \hat{Y}_{train} \setminus, \setminus, ,nbins=50$','Interpreter','latex')
axis([-1.8 1.8 0 125])

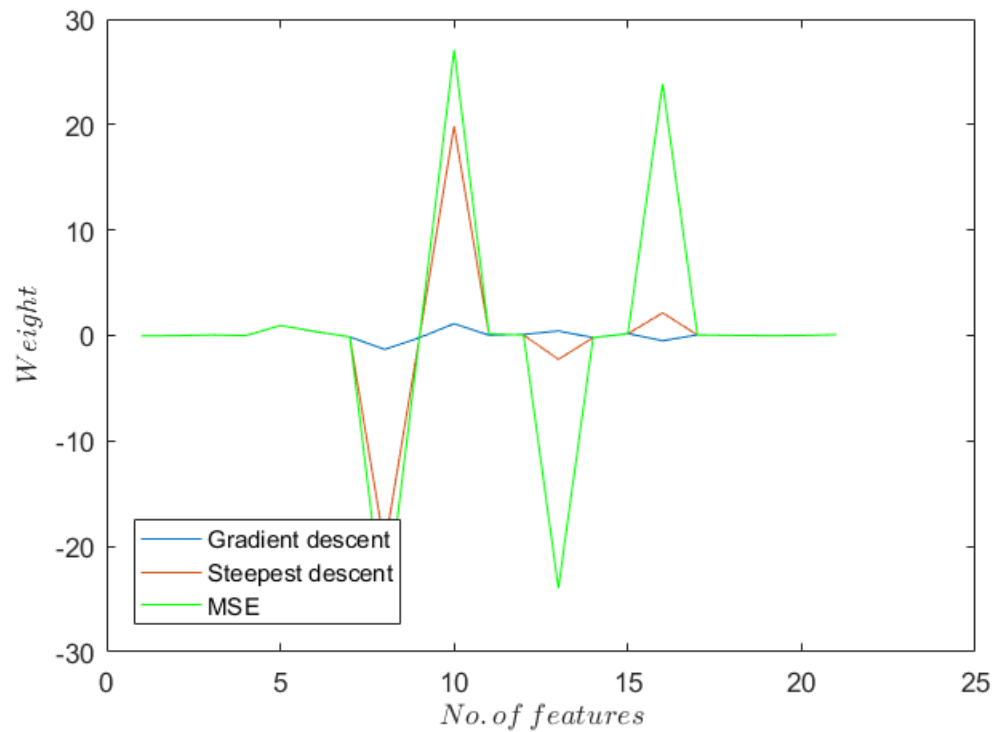
subplot(2,2,4)
histogram(Y_test-yhat_test_sd,nbins);
title('$Y_{test} \setminus, - \setminus, \hat{Y}_{test} \setminus, \setminus, ,nbins=50$','Interpreter','latex')
axis([-1.8 1.8 0 30])
suptitle('Steepest Descent Algorithm [F0 = 5]')

figure
xs = 1:length(a_mse);
plot(xs,a_grad(:,it_fin),xs,a_grad_sd(:,it_fin_sd),xs,a_mse,'g')
suptitle('Weights for each feature (a) [F0 = 5]')
xlabel('$No.\setminus, of\setminus, features$','Interpreter','latex')
ylabel('$Weight$','Interpreter','latex')
legend('Gradient descent','Steepest descent','MSE','Location','southwest')
```

Steepest Descent Algorithm [F0 = 5]



Weights for each feature (a) [F0 = 5]



Published with MATLAB® R2016b