

---

```
% Uncomment to add ll_ls to path
addpath('ll_ls_matlab');
clear;
clc;
close all;
```

**a)**

```
disp("Question 1 part A")
% Generating X and Y from a Uniform random variable
n = 500;
m = 200;
x_0 = 18;
lambda_set = [0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 2, 5, 10,
    15, 20, 30, 50, 100, 200, 600 ,800];

% Generating the sensing matrix
p=0.5;
A=(rand(200,500)<p);
A = 2*A/sqrt(m) - 1/sqrt(m);
A_t = A';

% Selecting the reconstruction and validation set sizes
R = 0.9*m;
V = 0.1*m;

% Generating the values of x from a Uniform Distribution of
% range(0,1000), and computing Y correspondingly
x = zeros(n,1);
indices = randperm(n, x_0);
x(indices) = randi([0,1000],x_0,1);
y = A*x;
sigma = 0.05*mean(abs(y));
y = y + (sigma.^(2)).*randn(m,1);

% Selecting R random indices from m to create the reconstruction set
arr_indices = 1:m;
R_indices = randperm(m,0.9*m);
R_y = y(R_indices);

% Considering the remaining as validation set
V_indices = setdiff(arr_indices, R_indices);
V_y = y(V_indices);

% Considering corresponding rows of A for the reconstruction and
% validations set
A_R = A(R_indices,:);
A_V = A(V_indices,:);

% Using the ll_ls solver, computing the estimated x from the
% reconstruction set for different values of lambda
```

---

```

validation_array = zeros(length(lambda_set),1);
rmse_array = zeros(length(lambda_set),1);
failed_set = 0;
for i=1:length(lambda_set)
    quiet = true;
    [x_estimated, status] = ll_ls(A_R, R_y, lambda_set(:,i), 0.01, quiet);
    if status == "Failed"
        failed_set = failed_set + 1;
    end
    validation_error = (V_y - A_V*x_estimated)'*(V_y - A_V*x_estimated)/
length(V_y);
    validation_array(i) = validation_error;
    difference = x_estimated - x;
    rmse = sqrt(difference'*difference)/sqrt(x'*x);
    rmse_array(i) = rmse;
end
disp(["Failed set length : " failed_set])

% Plotting the figure for lambda vs validation scores
figure(1)
plot(log(lambda_set), validation_array)
xticks(log(lambda_set))
xtickangle(90)
xlabel('log(\Lambda)')
ylabel('Val error')
title('Validation Error vs log(\Lambda)')
saveas(figure(1), '../output/val_vs_lambda.png')

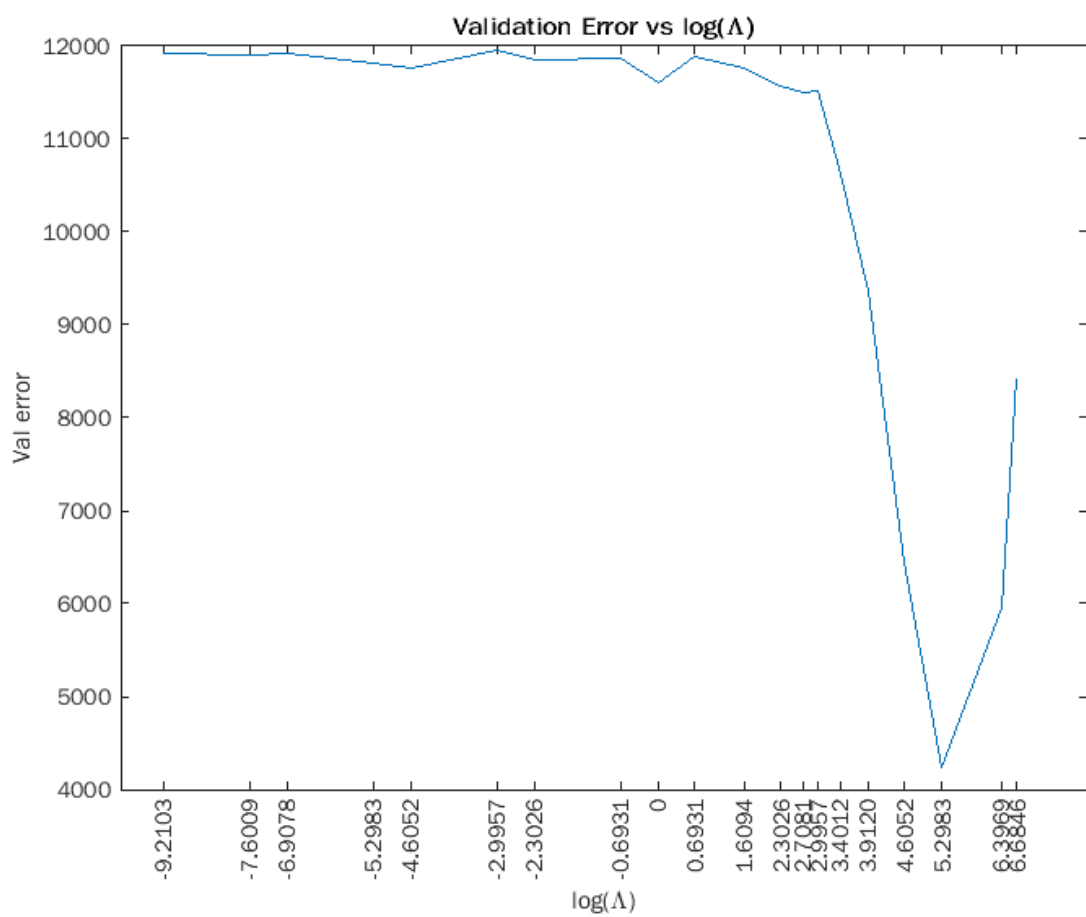
% Plotting the figure for lambda vs RMSE scores
figure(2)
plot(log(lambda_set), rmse_array)
xticks(log(lambda_set))
xtickangle(90)
xlabel('log(\Lambda)')
ylabel('RMSE')
title('RMSE vs log(\Lambda)')
saveas(figure(2), '../output/rmse_vs_lambda.png')

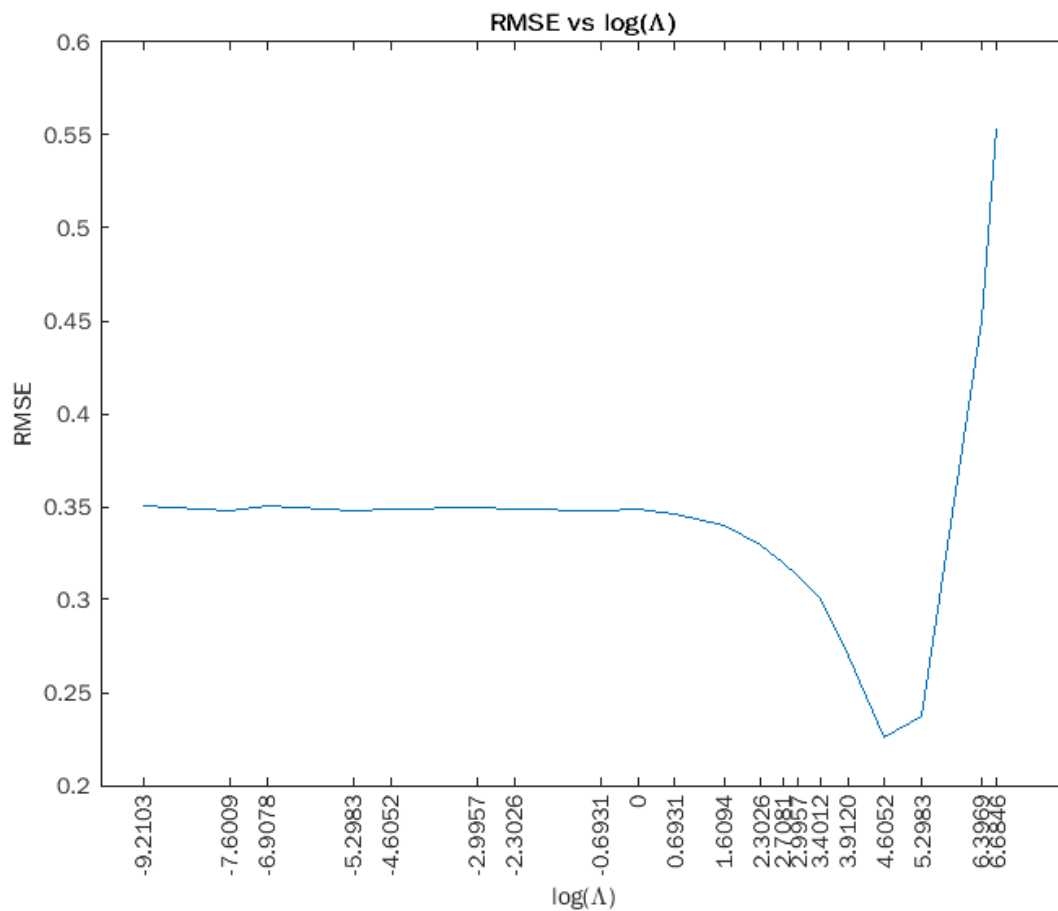
% Optimal value of lambda obtained from both the methods
[~,I1] = min(validation_array);
[~,I2] = min(rmse_array);
disp(["Optimal log lambda for minimum cross validation is",
    log(lambda_set(:,I1)), "and the corres. lambda value is", lambda_set(:,I1)])
disp(["Optimal log lambda for minimum rmse is", log(lambda_set(:,I2)), "and
    the corres. lambda value is", lambda_set(:,I2)])

```

Question 1 part A

"Failed set length : "	"0"		
"Optimal log lambda..."	"5.2983"	"and the corres. la..."	"200"
"Optimal log lambda..."	"4.6052"	"and the corres. la..."	"100"





## b) When V and R are not disjoint but coincident sets

```

disp("Quesiton 1 part b")
%%%%% NOTE : Keep Running this Block until we get solved status for all
           lambdas

R_y = y;
V_y = y;

% Considering corresponding rows of A for the reconstruction and
% validations set
A_R = A;
A_V = A;

% Using the l1_ls solver, computing the estimated x from the
% reconstruction set for different values of lambda
validation_array = zeros(17,1);
rmse_array = zeros(17,1);
failed_set = 0;
for i=1:length(lambda_set)

```

---

```

    quiet = true;
    [x_estimated, status] = ll_ls(A_R, R_y, lambda_set(:,i), 0.01, quiet);
    if status == "Failed"
        failed_set = failed_set + 1;
    end
    validation_error = (V_y - A_V*x_estimated)'*(V_y - A_V*x_estimated)/
length(V_y);
    validation_array(i) = validation_error;
    difference = x_estimated - x;
    rmse = sqrt(difference'*difference)/sqrt(x'*x);
    rmse_array(i) = rmse;
end

% Plotting the figure for lambda vs validation scores
figure(3)
plot(log(lambda_set), validation_array)
xticks(log(lambda_set))
xtickangle(90)
xlabel('log(\Lambda)')
ylabel('Val error')
title('Validation Error vs log(\Lambda) (Coincident Sets)')
saveas(figure(3), '../output/val_vs_lambda_coincident.png')

% Plotting the figure for lambda vs RMSE scores
figure(4)
plot(log(lambda_set), rmse_array)
xticks(log(lambda_set))
xtickangle(90)
xlabel('log(\Lambda)')
ylabel('RMSE')
title('RMSE vs log(\Lambda) (Coincident Sets)')
saveas(figure(4), '../output/rmse_vs_lambda_coincident.png')

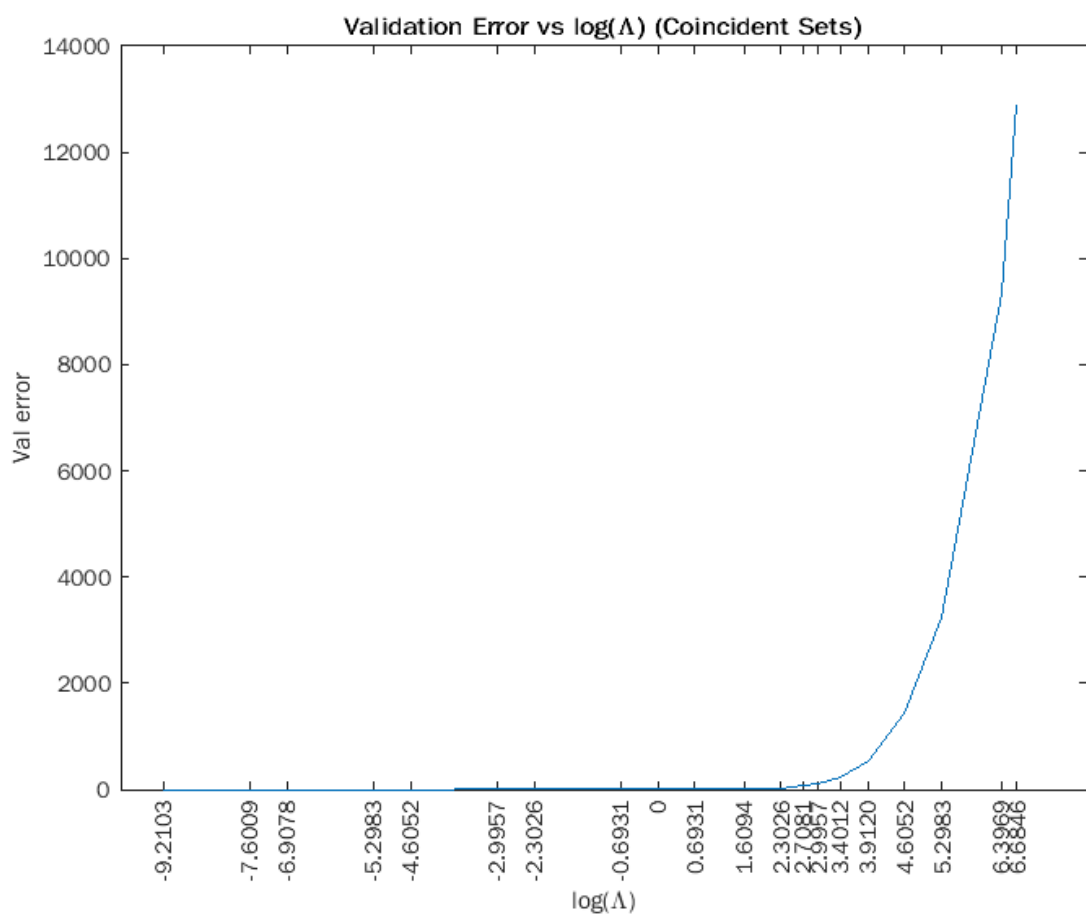
% Optimal value of lambda obtained from both the methods
[~,I1] = min(validation_array);
[~,I2] = min(rmse_array);
disp(["Optimal log lambda with coincident sets for minimum cross
validation is", log(lambda_set(:,I1)), "and the corres. lambda value is",
lambda_set(:,I1)])
disp(["Optimal log lambda with coincident sets for minimum rmse is",
log(lambda_set(:,I2)), "and the corres. lambda value is", lambda_set(:,I2)])

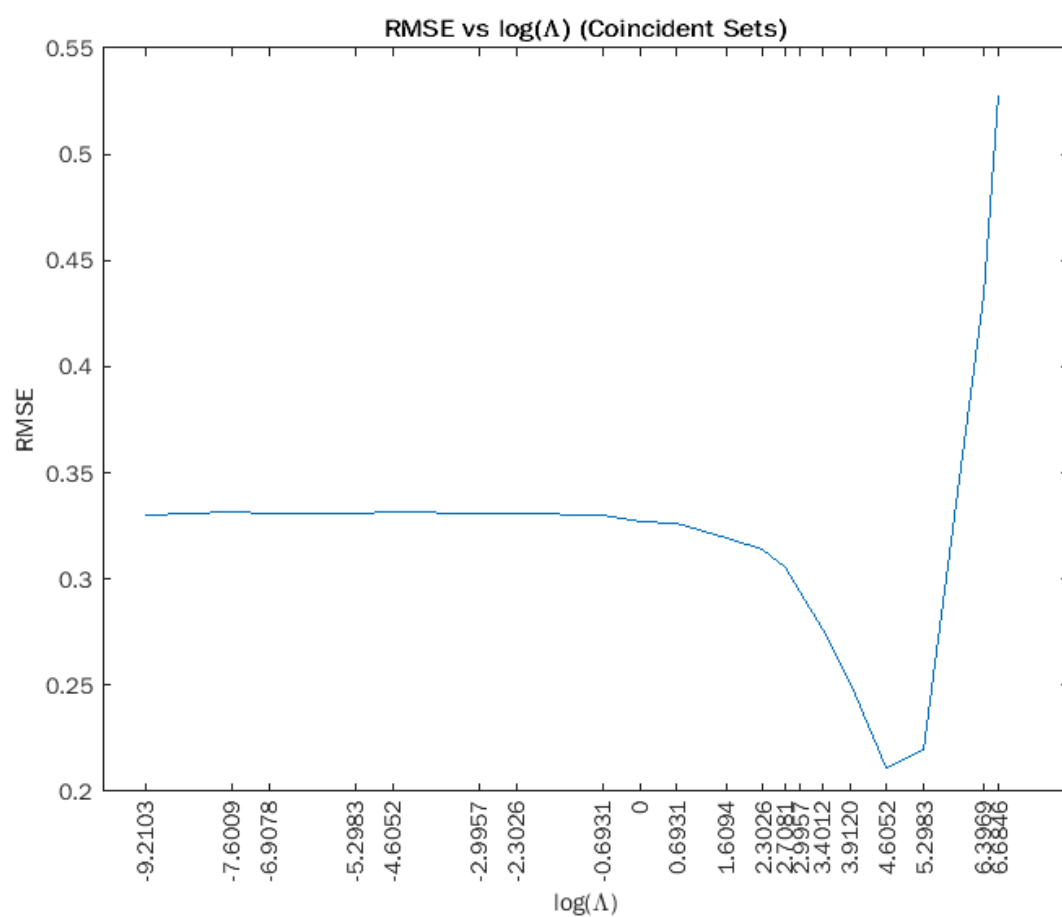
Quesiton 1 part b
    "Optimal log lambda..."    "-9.2103"    "and the corres. la..."    "0.0001"

    "Optimal log lambda..."    "4.6052"    "and the corres. la..."    "100"

```

---





*Published with MATLAB® R2022a*