APPC
TP 1 Lasso

Thomas ROBERT

Prepare data

```
data = load('housing.data');

% make X and y matrices
[n,d] = size(data);
p = d-1;
X = data(:, 1:p);
y = data(:,d);

% standardize feature values and center target
mu_y = mean(y);
y = y - mu_y;
[X, mu, sigma] = standardizeCols(X);

% Split learn and test
[Xlearn, ylearn, Xtest, ytest] = splitdata(X, y, 0.3);
```

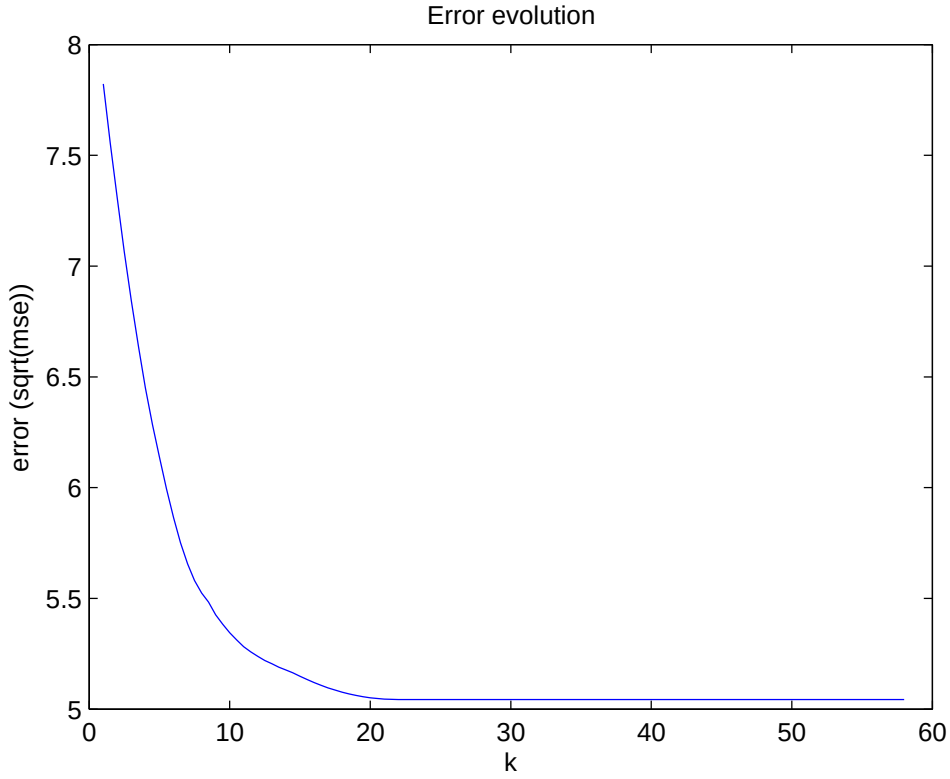
Solve the problem

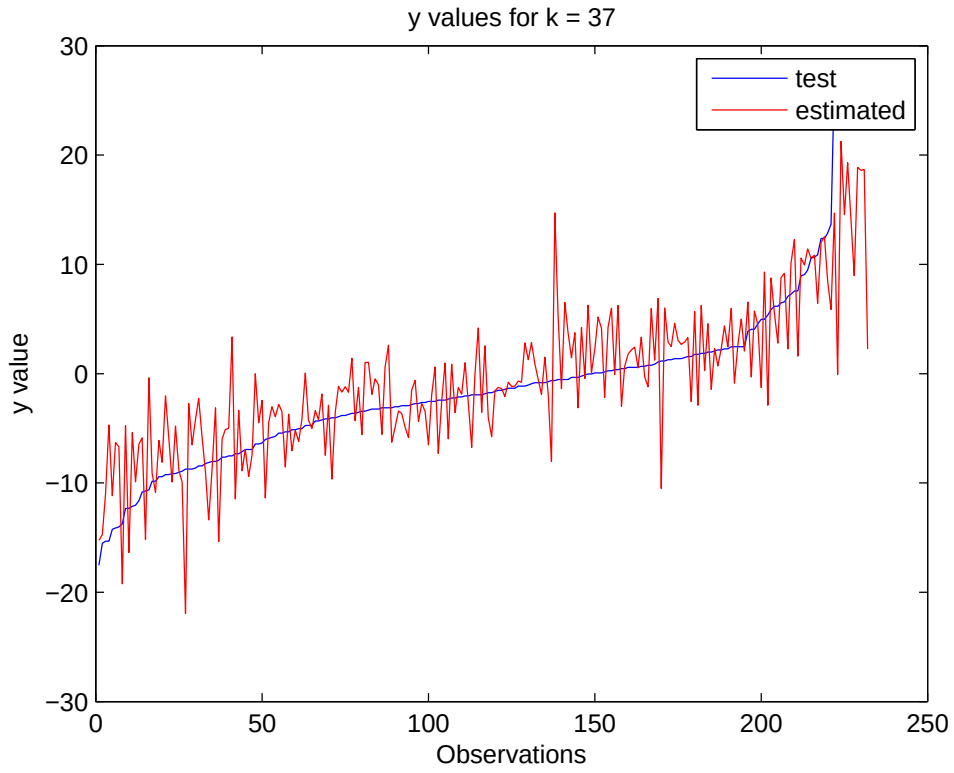
```
% test different values of k
kvals = [1:0.5:30 31:3:60];
errors = zeros(length(kvals), 1);
betas = zeros(length(kvals), p);

cvx_quiet(true);
for i = 1:length(kvals)
    k = kvals(i);

    % Resolve min problem
    cvx_begin
        % variables
        variables b(p)

        % objectif
        minimise(1/2 * b'*(Xlearn')*Xlearn*b - ylearn'*Xlearn*b)

        % contraintes
        subject to
            norm(b, 1) <= k
    cvx_end

    % Test
    betas(i, :) = b;
    ytest_hat = Xtest * b;
    errors(i) = sqrt(mean((ytest - ytest_hat).^2));
end
```

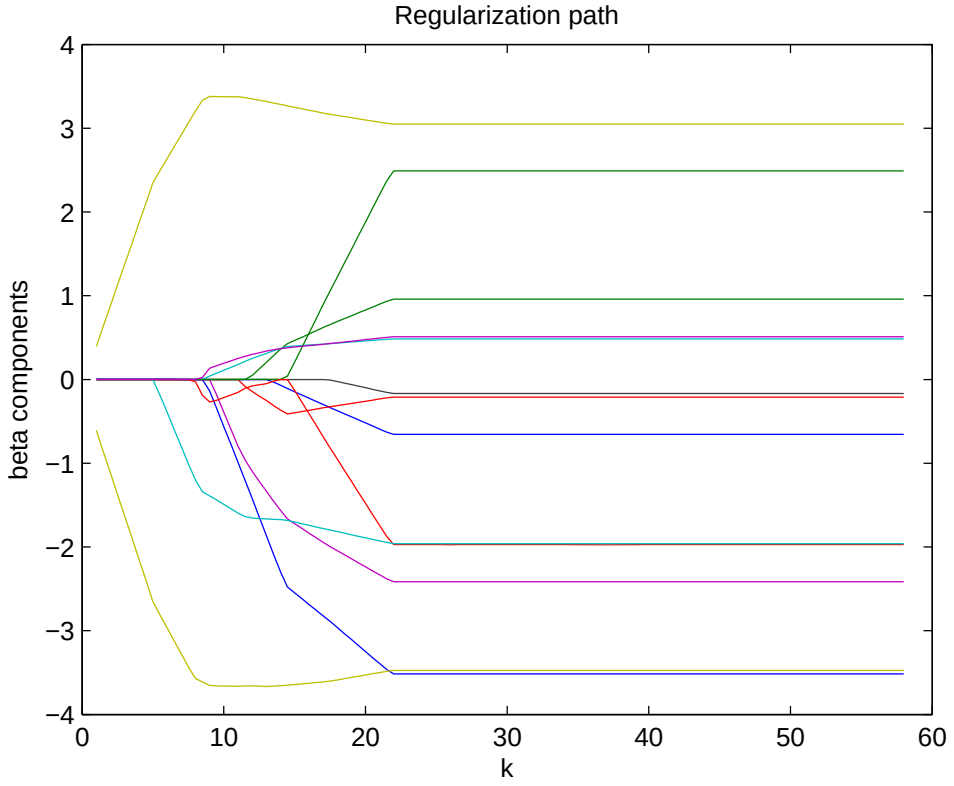
Plot results

```
% plot error evolution
figure;
plot(kvals, errors);
title('Error evolution');
xlabel('k');
ylabel('error (sqrt(mse))');

```

```
% plot best solution
[~, i] = min(errors);
k = kvals(i);
figure;
plot(ytest, 'b');
hold on;
plot(Xtest * betas(i, :)', 'r');
title(['y values for k = ' num2str(k)]);
xlabel('Observations');
ylabel('y value');
legend('test', 'estimated');

% plot regularization path
figure;
plot(kvals, betas);
title('Regularization path');
xlabel('k');
ylabel('beta components');
```







With monQP

```
H = [Xlearn'*Xlearn -Xlearn'*Xlearn
     -Xlearn'*Xlearn  Xlearn'*Xlearn];
c = [Xlearn'*ylearn
     -Xlearn'*ylearn];
A = ones(2*p,1);
l = 10^-12;
verbose = 0;
```

```
8
9  b = 10; % k
10
11 [xnew, lambda, pos] = monqp(H,c,A,b,inf,l,verbose);
12
13 Bpm = zeros(2*p,1);
14 Bpm(pos) = xnew;
15
16 beta = Bpm(1:p)-Bpm(p+1:end)
```

```
beta =

         0
         0
         0
    0.1102
   -0.3898
    3.3769
         0
   -0.5608
         0
   -0.2162
   -1.4917
    0.1936
   -3.6607
```