APPC
TP 2 Lasso Piecewise Regularization Path

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);
```

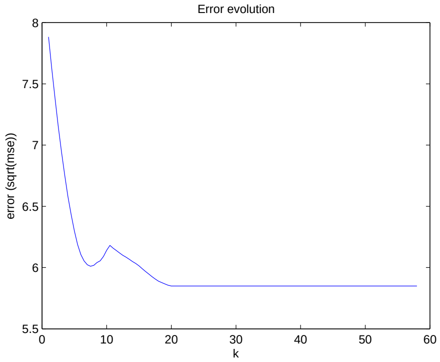
Solve the problem

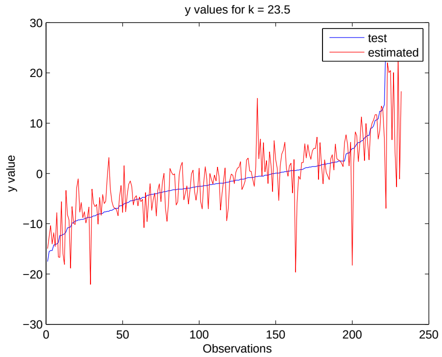
```
% test different values of k
kvals = [1:0.5:30 31:3:60];
errors = zeros(length(kvals), 1);
betas = zeros(length(kvals), p);

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
```

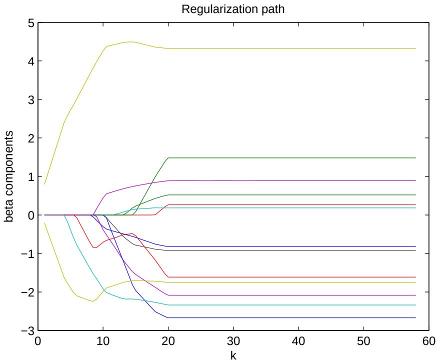
Plot results

```
% 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');
```

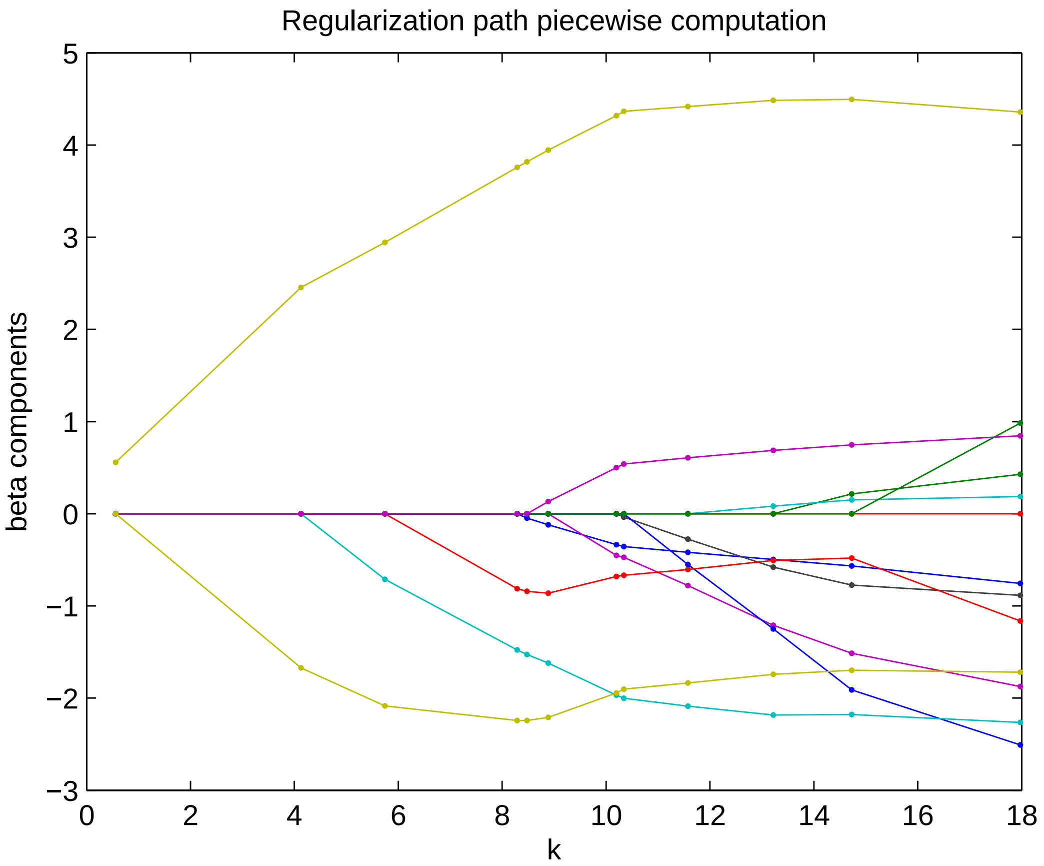






Piecewise computation of the regularization path

```
% compute some terms to simplify syntax
XX = (Xlearn'*Xlearn);

% Compute B0 = B MC
B0 = XX \ (Xlearn'*ylearn)

% Compute lambda1
v = XX\sign(B0)
lambda = B0./v
[lambdak, k] = lambdaMin(lambda)

% Compute B1
Bk =  B0 - lambdak * v

% Init IB
IB = setdiff(1:p,k)

% Loop
betas = [];
i = 0; % sup bound to be sure...
while(~isempty(IB) && i < 1000)

    % compute some terms to simplify syntax
    XIB = Xlearn(:,IB);
    XX = XIB' * XIB;
    v = XX \ sign(Bk(IB));

    % Compute lambdak+1 et k+1
    lambda = ones(p, 1) * Inf;
    lambda(IB) = (Bk(IB) + lambdak * v) ./ v;
    [lambdakp1, kp1] = lambdaMin(lambda);

    % Compute Bk+1
    Bkp1 = zeros(p, 1);
    Bkp1(IB) = Bk(IB) - (lambdakp1 - lambdak)* v;

    % Rewrite names for next iteration
    betas = [betas Bk];
    Bk = Bkp1;
    lambdak = lambdakp1;
    IB = setdiff(IB,kp1);
end

% Plot resuts

ks = sum(abs(betas));
figure;
plot(ks, betas', '.-');
title('Regularization path piecewise computation');
xlabel('k');
ylabel('beta components');
```


Regularization path piecewise computation
beta components
k