APPC
TP 4 Adaptative Lasso

Thomas Robert

Prepare data

First we prepare the data : we standardize the data and create 2 datasets for learning and testing.

```
clear all;
close all;
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, 0.5);

% Rename elements because I'm lazy
X = Xlearn;
y = ylearn;
n = length(y);
```

Compute lasso and compare methods

```
% Hyperparameter

lambda = 10;

% Standard lasso

bLasso = CWLasso(X, y, lambda, zeros(p,1));

% Primal Adapatative Lasso with CVX

w = 1./abs((X'*X)\(X'*y));

cvx_quiet(true);
cvx_begin
    variables b(p)
    minimise(1/2 * sum_square(y - X * b) + lambda * w' * abs(b))
cvx_end

bPrimCVX = b;

% Primal Adapatative Lasso with monQP

k = w'*abs(bPrimCVX);

A = [w
     w];
c = [ X'*y
     -X'*y];
```

```
b = k;
H = [ X'*X -X'*X
     -X'*X  X'*X];
C = ones(2*p,1)*Inf;

[bvals, ~, pos] = monqp(H, c, A, b, C, 1e-6, false);
b = zeros(2*p, 1);
b(pos) = bvals;
b = b(1:p) - b(p+1:end);

bPrimQP = b;

% Dual Adaptative Lasso with CVX

cvx_begin
    variables a(n)
    minimise(1/2 * sum_square(a) - a'*y)
    subject to
        abs(X'*a) <= lambda*w
cvx_end

bDualCVX1 = (X'*X)\(X'*(y - a))

% Dual Adaptative Lasso with CVX v2

cvx_begin
    variables b(p)
    minimise(1/2 * sum_square(X * b))
    subject to
        abs(X'*(y-X*b)) <= lambda*w
cvx_end

bDualCVX2 = b

% Plot results

[bLasso bPrimCVX bPrimQP bDualCVX1 bDualCVX2]
```

```
bDualCVX1 =

   -0.8697
    0.5467
   -0.0000
    0.3408
   -2.2470
    3.1058
   -0.0836
   -2.6849
    2.2315
   -1.8133
   -2.5908
    0.5475
   -3.0870


bDualCVX2 =

   -0.8697
    0.5467
   -0.0000
    0.3408
   -2.2470
    3.1058
   -0.0836
   -2.6849
    2.2315
   -1.8133
   -2.5908
    0.5475
   -3.0870


ans =

   -0.8487   -0.8697   -0.8697   -0.8697   -0.8697
    0.5366    0.5468    0.5468    0.5467    0.5467
         0    0.0000         0   -0.0000   -0.0000
    0.3949    0.3408    0.3408    0.3408    0.3408
   -2.0879   -2.2471   -2.2470   -2.2470   -2.2470
    3.1592    3.1058    3.1058    3.1058    3.1058
   -0.2832   -0.0836   -0.0837   -0.0836   -0.0836
   -2.6463   -2.6850   -2.6850   -2.6849   -2.6849
    1.9652    2.2317    2.2316    2.2315    2.2315
   -1.6059   -1.8134   -1.8133   -1.8133   -1.8133
   -2.5257   -2.5908   -2.5908   -2.5908   -2.5908
    0.5734    0.5476    0.5475    0.5475    0.5475
   -2.9994   -3.0870   -3.0870   -3.0870   -3.0870
```