CIS 515

Fundamentals of Linear Algebra and Optimization
Jean Gallier

Project 5: Ridge Regression

The purpose of this project is to implement versions of ridge regression.

Recall that ridge regression for learning an affine function $f(x) = x^\top w + b$ from the training data $((x_1, y_1), \ldots, (x_m, y_m))$ is the following optimization problem:

Program (RR3):

$$\begin{aligned} &\text{minimize} \quad \xi^\top \xi + K w^\top w \\ &\text{subject to} \\ &\qquad\qquad y - Xw - b\mathbf{1}_m = \xi, \end{aligned}$$

with $y, \xi, \mathbf{1}_m \in \mathbb{R}^m$ and $w \in \mathbb{R}^n$ ($\mathbf{1}_m$ is the vector (of dim m) whose components are all equal to 1) and $K > 0$ a fixed constant.

Here X is an $m \times n$ matrix whose rows are the transpose of the data points $x_1, \ldots, x_m \in \mathbb{R}^n$, and $y \in \mathbb{R}^m$.

The first solution is obtained by centering the data: the centered data are $\widehat{y} = y - \overline{y}\mathbf{1}_m$ and $\widehat{X} = X - \overline{X}$, where $\overline{X}$ is the $m \times n$ matrix whose jth column is $\overline{X^j}\mathbf{1}_m$, the vector whose coordinates are all equal to the mean $\overline{X^j}$ of the jth column X^j of X.

The optimal solution w is given by

$$w = \widehat{X}^\top (\widehat{X}\widehat{X}^\top + KI_m)^{-1}\widehat{y}, \qquad (*_{w6})$$

and b is given by

$$b = \overline{y} - (\overline{X^1} \cdots \overline{X^n})w,$$

where $(\overline{X^1} \cdots \overline{X^n})$ is the $1 \times n$ row vector consisting the the means of the columns of X.

(1) (**20 points**) Write a `Matlab` function `ridgeregv1` to compute w and b from X and y. This function takes X, y, and $K > 0$ as input, and returns w, the Euclidean norm nw of w, b, the error vector $xi = \widehat{y} - \widehat{X}w$, and the Euclidean norm nxi of xi.

```
function [w,nw,b,xi,nxi] = ridgeregv1(X,y,K)
%  Ridge regression with centered data
%  b is not penalized
%  X is an m x n matrix, y a m x 1 column vector
%  weight vector w, intercept b
```

```
%  Solution in terms of the primal variables
%
m = size(y,1);
n = size(X,2);
%
% Your code
%
end
```

(2) (**20 points**) The dual of Program (**RR3**) is

Program (DRR3):

$$\begin{aligned} &\text{minimize} \quad \alpha^\top(XX^\top + KI_m)\alpha - 2\alpha^\top y \\ &\text{subject to} \\ &\qquad\qquad \mathbf{1}^\top\alpha = 0, \end{aligned}$$

where the minimization is over α. This program can be solved directly without centering the data by solving the KKT equations

$$\begin{pmatrix} XX^\top + KI_m & \mathbf{1}_m \\ \mathbf{1}_m^\top & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \mu \end{pmatrix} = \begin{pmatrix} y \\ 0 \end{pmatrix}.$$

Then we have

$$\begin{aligned} w &= X^\top\alpha \\ b &= \mu \\ \xi &= K\alpha. \end{aligned}$$

Write a `Matlab` function `ridgeregb1` to compute w, b, α and ξ from X and y. This function takes X, y, and $K > 0$ as input, and returns w, b, the error vector xi, the Euclidean norm nxi of xi, and α.

```
function [w,b,xi,nxi,alpha] = ridgeregb1(X,y,K)
%  Ridge regression
%  b is not penalized
%  X is an m x n matrix, y a m x 1 columm vector
%  weight vector w, intercept b
%  Solution in terms of the KKT equations
%
m = size(y,1);
n = size(X,2);
X1 = X*X' + K*eye(m);
%
```

```
% Your code
%
end
```

Compare the solutions for w and b given by `ridgerev1` and `ridgeregb1` (they should agree up to roughly ten decimals).

(3) (**20 points**) Another way to solve ridge regression is to penalize b. This corresponds to the following optimization problem:

$$\begin{aligned} &\text{minimize} \quad \xi^\top \xi + K w^\top w + K b^2 \\ &\text{subject to} \\ &\qquad\qquad y - Xw - b\mathbf{1}_m = \xi, \end{aligned}$$

minimizing over ξ, w and b.

This suggests treating b an an extra component of the weight vector w and by forming the $m \times (n+1)$ matrix $[X\ \mathbf{1}]$ obtained by adding a column of 1's (of dimension m) to the matrix X, and we obtain

Program (RR3*b*):

$$\begin{aligned} &\text{minimize} \quad \xi^\top \xi + K w^\top w + K b^2 \\ &\text{subject to} \\ &\qquad\qquad y - [X\ \mathbf{1}] \begin{pmatrix} w \\ b \end{pmatrix} = \xi, \end{aligned}$$

minimizing over ξ, w and b.

It can be shown that the solution is given by

$$\begin{aligned} \alpha &= ([X\ \mathbf{1}][X\ \mathbf{1}]^\top + K I_m)^{-1} y \\ \begin{pmatrix} w \\ b \end{pmatrix} &= [X\ \mathbf{1}]^\top \alpha \\ \xi &= K\alpha. \end{aligned}$$

Thus $b = \mathbf{1}^\top \alpha$.

Write a `Matlab` function `ridgeregv2` to compute w and b from X and y. This function takes X, y, and $K > 0$ as input, and returns w, the Euclidean norm nw of w, b, the error vector $xi = K\alpha$, and the Euclidean norm nxi of xi.

```
function [w,nw,b,xi,nxi] = ridgeregv2(X,y,K)
%  Ridge regression minimizing w and b
%  b is penalized
```

```
%  X is an m x n matrix, y a m x 1 column vector
%  weight vector w, intercept b
%  Solution in terms of the dual variable
%
m = size(y,1); n = size(X,2);
XX = [X ones(m,1)];
%
% Your code
%
end
```

(4) (**20 points**) As a least squares problem, the solution is given in terms of the pseudo-inverse $[X\ \mathbf{1}]^+$ of $[X\ \mathbf{1}]$ by

$$\begin{pmatrix} w \\ b \end{pmatrix} = [X\ \mathbf{1}]^+ y.$$

Write a `Matlab` function `reglq` to compute w and b from X and y. This function takes X, y as input, and returns w, the Euclidean norm nw of w, b, the error vector $xi = y - Xw - b\mathbf{1}$, and the Euclidean norm nxi of xi.

```
function [w,nw,b,xi,nxi] = reglq(X,y)
%  Regression minimizing w and b
%  X is an m x n matrix, y a m x 1 column vector
%  weight vector w, intercept b
%  Computes the least squares solution using the pseudo inverse
%  Use pin
%
m = size(y,1); n = size(X,2);
XX = [X ones(m,1)];
%
% Your code
%
end
```

(5) (**20 points**) To test your four functions, run the following `Matlab` function `reg3`:

```
function [w1,w2,w3,w4] = reg3(X,y,K)
%  Calls four regression methods
%  Ridge regression minimizing w, b, not penalizing b
%  Ridge regression minimizing w, b, not penalizing b,
%  using the KKT eqs
```

```
%  Ridge regression minimizing w, b, penalizing b
%  Least squares, penalizing b
%  X is an m x n matrix, y a m x 1 column vector
%  weight vector w, intercept b
%
m = size(y,1); n = size(X,2);
[w1,nw1,b1,~,~] = ridgeregv1(X,y,K);
[w2,b2,xi2,nxi2,alpha] = ridgeregb1(X,y,K);
[w3,nw3,b3,~,~] = ridgeregv2(X,y,K);
[w4,nw4,b4,~,~] = reglq(X,y);
fprintf('b1 =  %.15f \n',b1)
fprintf('b2 =  %.15f \n',b2)
fprintf('b3 =  %.15f \n',b3)
fprintf('b4 =  %.15f \n',b4)

if n == 1
      [ll,mm] = showgraph(X,y);
      hold off
      [ll,mm] = showgraph(X,y);
      ww1 = [w1;-1]; ww3 = [w3;-1];
      ww4 = [w4;-1];
      n1 = sqrt(ww1'*ww1); n3 = sqrt(ww3'*ww3);
      n4 = sqrt(ww4'*ww4);
      l1 = makeline(ww1,-b1,ll,mm,n1);     % best fit, ridge 1
      l2 = makeline(ww3,-b3,ll,mm,n3);     % best fit,
                                           % ridge penalizing b
      l3 = makeline(ww4,-b4,ll,mm,n4);     % best fit, least squares
      plot(l1(1,:),l1(2,:),'-m','LineWidth',1.2)  % magenta best
      plot(l2(1,:),l2(2,:),'-r','LineWidth',1.2)  % red
      plot(l3(1,:),l3(2,:),'-b','LineWidth',1.2)  % blue
      hold off
 else
     if n == 2
         offset = 5;
         [ll,mm] = showpoints(X,y,offset);
         axis equal
         axis([ll(1) mm(1) ll(2) mm(2)]);
         view([-1 -1 1]);
         xlabel('X','fontsize',14);ylabel('Y','fontsize',14);
         zlabel('Z','fontsize',14);
         hold off
         [ll,mm] = showpoints(X,y,offset);
```

```
        C3 = [0 0 1]; % blue
        C1 = [1 0 1]; % magenta
        C2 = [1 0 0]; % red
        plotplane(w1,b1,ll,mm,C1)    % best fit, ridge 1, magenta
        plotplane(w3,b3,ll,mm,C2)    % best fit, ridge
                                     % penalizinbg b, red
        plotplane(w4,b4,ll,mm,C3)    % best fit, least squares, blue
        axis equal
        axis([ll(1) mm(1) ll(2) mm(2)]);
        view([-1 -1 1]);
        xlabel('X','fontsize',14);ylabel('Y','fontsize',14);
                zlabel('Z','fontsize',14);
        hold off
    end
 end
end
```

The functions `showgraph, makeline, showpoints, plotplane` are in the folder Matlabcode5.

To ensure that we can check your results, it is *crucial that you set the seed of the random number generator by using the command*

```
rng(97922758)
```

Run `reg3` for $K = 0.01,\ 0.1,\ 1,\ 10,\ 100,\ 1000$ on the following data sets:

$$X3 = [-10; -8; -6; -4; -2; 2; 4; 6; 8; 10; -9; -7; -5; -3; -1; 1; 3; 5; 7; 9]$$
$$y3 = [-3; 1; 0; 0; 1.5; 4; 6; 5; 1; 8; -2.5; 0.5; 1.5; -1; -0.5; 3.5; 5.5; 2.5; 4.5; 5]$$

```
X13 = 15*randn(50,1);
ww13 = 1;
y13 = X13*ww13 + 10*randn(50,1) + 20;
```

$$X = [-10\ 11; -6\ 5; -2\ 4; 0\ 0; 1\ 2; 2\ -5; 6\ -4; 10\ -6]$$
$$y = [0; -2.5; 0.5; -2; 2.5; -4.2; 1; 4]$$

```
X8 = 10*randn(50,2);
ww = [1; 2];
y8 = X8*ww + 10*randn(50,1) + 10;
```

```
X10 = 10*randn(100,2);
ww2 = [1; 2];
y10 = X10*ww2 + 10*randn(100,1) + 15;

X20 = randn(50,30);
ww20 = [0; 2; 0; -3; 0; -4; 1; 0; 2; 0; 2; 3; 0; -5; 6; 0; 1; 2; 0;
        10; 0; 0; 3; 4; 5; 0; 0; -6; -8;0];
y20 = X20*ww20 + randn(50,1)*0.1 + 5;
```