Contents

Warm Up

```
rng(19875)
X = randi([0, 1], [1000 5000]);
X_train = X(1:900,:);
X_test = X(901:1000,:);
```

1)

Solution: a. Base on the sparsity of the weights, LASSO would be a good candicate to learning the coeeficients.

b. The MSE for LASSO is much smaller than the ones for Ridge. Because we have a sparse coefficient, LASSO's solution is closer to the true weight, while Ridges would assign weights to unsignificant features.

```
% data preparation:
rng(19875)
w = zeros(5000,1);
w(1:15,:) = 1;
y = X*w+randi([-1,1],1000,1);
y_train = y(1:900,:);
y_test = y(901:1000,:);
```

```
[L,LFitInfo] = lasso(X_train,y_train,'Alpha',1);
[M,I] = min(LFitInfo.MSE);
e= (1/100)*norm(X_test*L(:,I)-y_test)^2
[R,RFitInfo] = lasso(X_train,y_train,'Alpha',10e-100);
[M,I] = min(RFitInfo.MSE);
e= (1/100)*norm(X_test*R(:,I)-y_test)^2
```

```
e =

    0.4630


e =
```

```
    59.3900
```

2)

Solution: a. Since the weight is not as sparse as the previous one, Ridge would be a good candidate for learning the weights for this model.

b. The MSE for Ridge is much smaller than the ones for LASSO. Because we have coefficients that are not very sparse, using Ridge will help us to find closer weights to true values and reduce noise amplification.

```
rng(19875)
w = randi([-1,1],5000,1);
w(1:500,:) = 0;
y = X*w+randi([-1,1],1000,1);
y_train = y(1:900,:);
y_test = y(901:1000,:);
```

```
[L,LFitInfo] = lasso(X_train,y_train,'Alpha',1);
[M,I] = min(LFitInfo.MSE);
e= (1/100)*norm(X_test*L(:,I)-y_test)^2 % MSE for LASSO
[R,RFitInfo] = lasso(X_train,y_train,'Alpha',10e-100);
[M,I] = min(RFitInfo.MSE);
e= (1/100)*norm(X_test*R(:,I)-y_test)^2 % MSE for Ridge
```

```
e =

   1.1002e+03


e =

   1.0267e+03
```

3)

Solution: Given strong prior knowledge of the percentage of important features in a dataset, it makes sense to choose LASSO or Ridge in place of Elastic Net. However, in absence of prior knowledge, Elastic Net should be the preferred solution. Elastic Net also has the advantage of taking care of highly correlated features which often appear in huge data set. It picks out important feature groups and average the weights between the features in each group, which reduces the randomness of LASSO.

Main Activity

read in data from csv. X stores data from 72 patients with 3571 features classified as two classes. Among which, we use 1 to represent calss "ALL" and -1 tp represent "AML"

```
data = csvread('leukemia_small.csv', 0, 0);
y =data(1,:)';
X =data(2:3572,:)';
```

1)

Solution: Using LASSO Code to find the desirable coefficients

```
[L,LFitInfo] = lasso(X,y,'Alpha',1);
% a) MSE for LASSO
[M,I] = min(LFitInfo.MSE);
MSE_LASSO = M

% b) Since lambda_1 = lambda * Alpha
lambda_1 = 1 * LFitInfo.Lambda(I)

% c) number of nonzero weights
no_nonzero = LFitInfo.DF(I)
```

```
MSE_LASSO =

   8.7164e-04


lambda_1 =

    0.0071


no_nonzero =

    71
```

2)

Solution: Using Ridge Regression Code to find the desirable coefficients

```
[R,RFitInfo] = lasso(X,y,'Alpha',10e-10);
% a) MSE for Ridge
[M,I] = min(RFitInfo.MSE);
MSE_RIDGE = M

% b) Since lambda_2 = lambda * (1-Alpha)/2
lambda_2 = (1 - 10e-10)/2 * RFitInfo.Lambda(I)

% c) number of nonzero weights
no_nonzero = RFitInfo.DF(I)
```

```
MSE_RIDGE =

    0.9008


lambda_2 =

   4.0931e+04


no_nonzero =

        3571
```

3)

MSE is much smaller if we use LASSO for regulization. It can be an indication that there are many unrelated features to prediction as well.

4)

lambda_1 = lambda * Alpha

lambda_2 = lambda * (1-Alpha)/2

lambda_1 / lambda_2 = 2 Alpha / (1 - Alpha)

2 Alpha * lambda_2 = lambda_1 - lambda_1 * Alpha

Alpha = lambda_1 /(lambda_1 + 2 * lambda_2)

```
Alpha = lambda_1 /(lambda_1 + 2 * lambda_2)
```

```
Alpha =

   8.6975e-08
```

5)

We have tried these Alphas, you might find that your best alpha might have better or worse MSE than our best one. Yes, it is the smallest.

```
alpha = [10e-10,10e-9,10e-8,10e-7,10e-6,10e-5,10e-4,10e-3,10e-2,10e-1];
m = intmax;
MSE_arr = zeros(10);
for i=1:10
   [E,EFitInfo] = lasso(X,y,'Alpha',alpha(i));
   [M,I] = min(EFitInfo.MSE);
   MSE_arr(i) = M;
   if (M< m)
        m = M;
        weights = E(:,I);
        non_zero = EFitInfo.DF(I);
   end
end
% a) MSE for Elastic Net
MSE_Elastic = m

% b)
non_zero
```

```
MSE_Elastic =

   7.7592e-04


non_zero =
```

```
        3386
```

c) Rank <= n<p. Suppose every features are relevant to the regression. LASSO pick one feature per group of highly correlated features. So the total number of features picked by LASSO <= rank<= n<p. Elastic Net only filters out some groups of features which doesn?t have constrain on the total number of features picked.

d) There are only 3571 - 3386 = 185 features not used in Elastic Net model, which does not follow the assumption in quesiton 3. In fact, this can be an indication that a samll portion of features are irrelavent for prediction. Among the relevent features, many of them can be grouped with other highly correlated features, while LASSO only chose one of them from each group to constructe a sparse weight.

6)

Solution:

iv. Features 6, 7, 8 are highly correlate but they could be trivial features which do not contribute to the classification and be assinged 0 weight. Feature 9 10, 11, 12 are high correlated and they are important to the classification model. They are picked out and assigned roughly similar weights by the Elastic Net.

```
[E,EFitInfo] = lasso(X,y,'Alpha',0.001);
[M,I] = min(EFitInfo.MSE);
C =  corrcoef(X);
temp = C(1:12,1:12);
weight = E(:,I);
```

Published with MATLAB® R2016a