Project III —Feature Selection

Group 18
Jiapeng Sun
Hao Xu
2021/6/5

Part 1 –
Prediction quality vs feature selection

Task 1 – Simulate data

Here, we set 7 values for the number of features and 6 values for sparsity; for each n/s combination, we simulate 10 times. As a result, we get 7*6*10 = 420 simulated datasets.

Number of features = [200, 300, 400, 500, 600, 700, 800]

Sparsity = [0.75, 0.80, 0.85, 0.9, 0.95, 0.99]

Task 2 - Determine λmin and λ1se

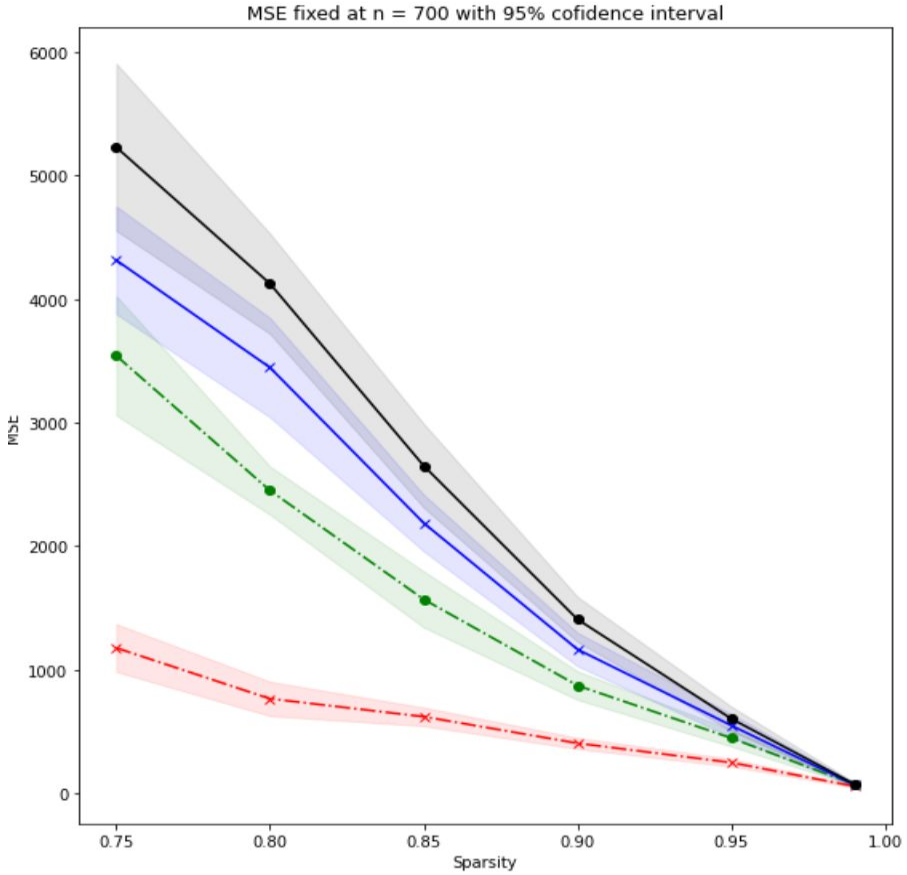
- Split each dataset into train and test set;
- Fit LassoCV function to on the train set and determine λmin and λ1se from the result;
- Fit Lasso on the train set using λmin and λ1se and calculate the predicted y on train and test sets;
- Calculate the MSE for both λmin and λ1se on both train and test set ;
- Encoded the weights from original data and selection from model into binary codes.
- Calculate the sensitivity and specificity on feature selection for both λmin and λ1se models

Task 3 - Comparing two Lasso models

- For each n/s combination, we take the average value for each performance metric on the 10 repeated results;
- When doing visualization, we also draw the 95% confidence interval for each metric.

Visualize MSE on Sparsity

When visualizing the MSE on sparsity for λmin and λ1se, we fixed 700 as number of features to avoid optimum n value, since it may leads to low difference on MSE for different s value.

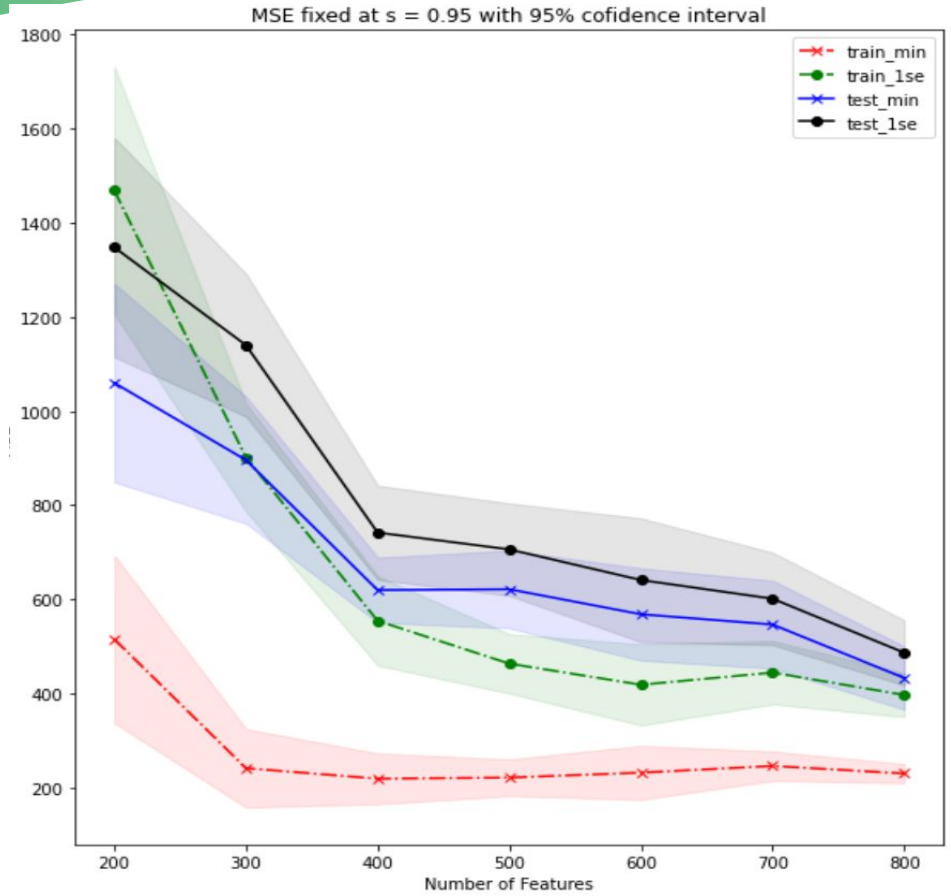


Visualize MSE on Sparsity

- Compare with model based on λ1se, the one with λmin shows better performance since more features are selected.
- As sparsity increases:
    1. The MSE shows gentle decrease on train set.
    2. The MSE shows sharp decrease on test set.
    3. The gap between MSE shows decrease on both train and test set.
- The reason why higher sparsity leads to better performance, may because that it leads to a more simple correlation as more weights are set to zero, which can be captured by the model easier as the result.

Visualize MSE on Number of Features

When visualizing the MSE on different number of features, we fixed 0.99 as value of sparsity, to avoid optimum s value, since it may leads to low difference on MSE for different n value.

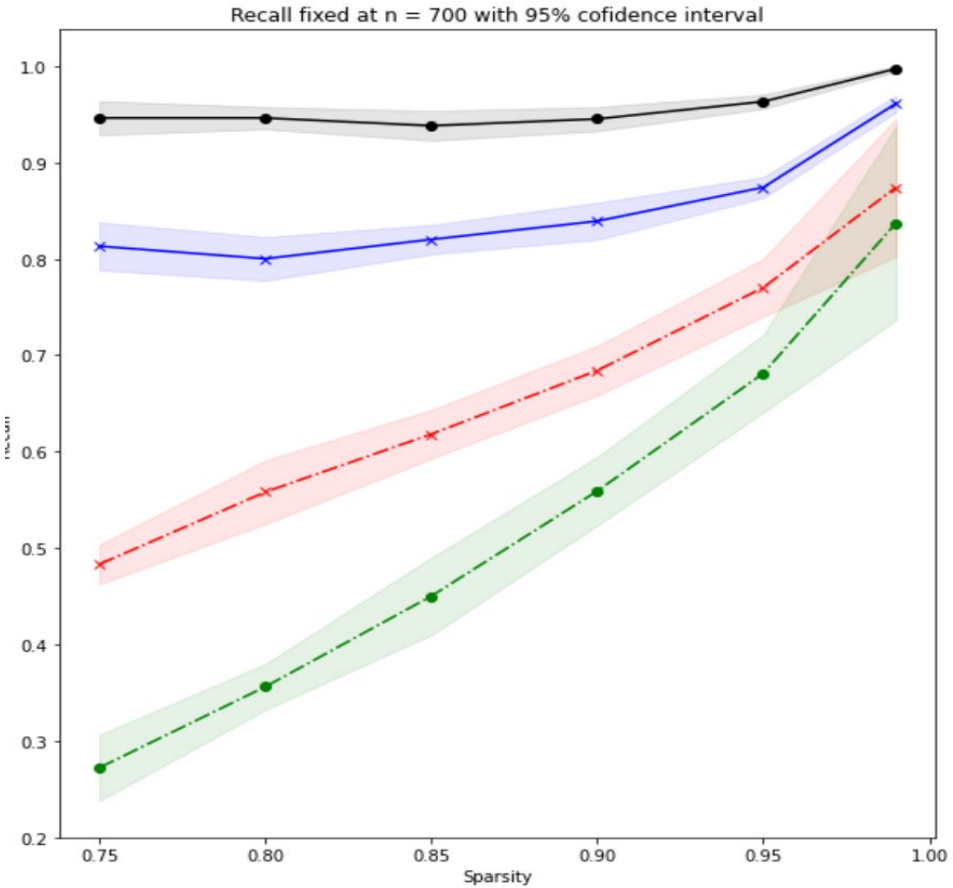


Visualize MSE on Number of Features

- Compare with model based on λ1se, the one with λmin shows better performance since more features are selected.
- As number of features increases:
    1. The MSE shows unstable decrease trend on both train and test set.
    2. The gap between MSE shows decrease on both train and test set.
- The reason why higher number of features leads to better performance, may because that it leads a model with more correlations, which can reduce the effect of mis-selected features.

Visualize Sensitivity and Specificity on Sparsity

When visualizing the Recall on sparsity for λmin and λ1se, we fixed 700 as number of features to avoid optimum n value, since it may leads to low difference on MSE for different s value.

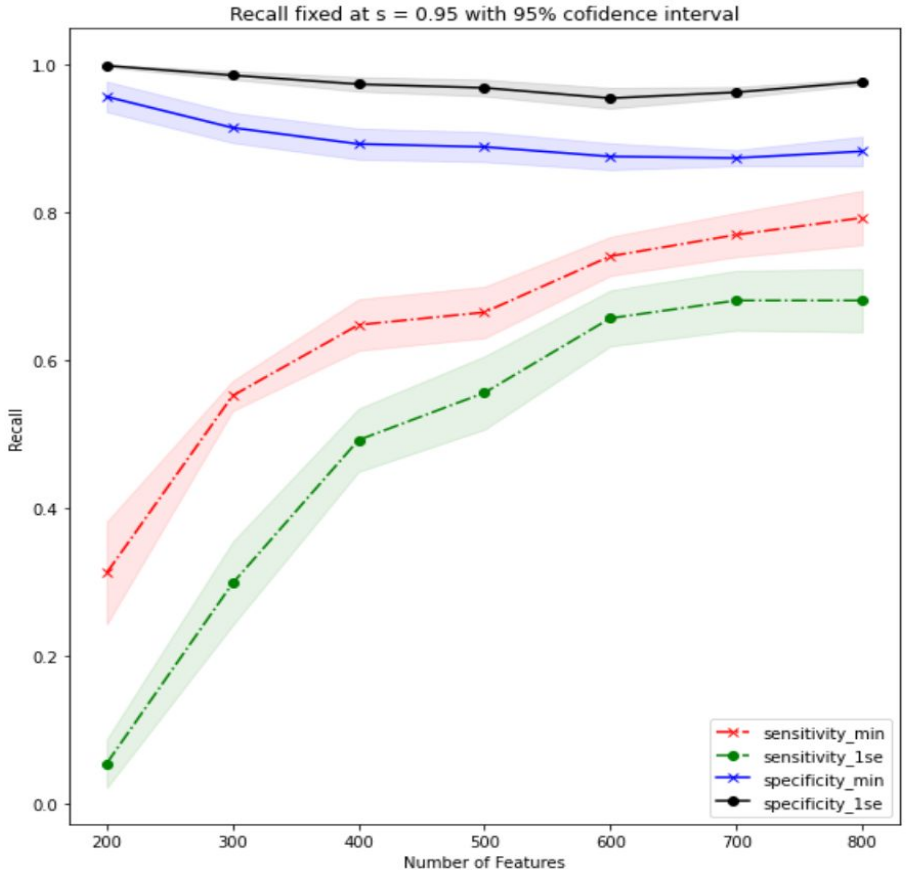


Visualize Sensitivity and Specificity on Sparsity

- Compare with model based on λ1se, the one with λmin shows:
    1. Lower sensitivity since less features are selected.
    2. Higher specificity since less features are selected.
- As sparsity increases, the gap between Recalls on λmin and λ1se shows decrease on both sensitivity and specificity.
- The reason why higher sparsity leads lower gap, may because it leads to a more simple correlation as more weights are set to zero, which can be captured by the model easier.

Visualize Sensitivity and Specificity On Number of Features

When visualizing the Recall on different number of features, we fixed 0.99 as value of sparsity, to avoid optimum s value, since it may leads to low difference on MSE for different n value.



Visualize Sensitivity and Specificity On Number of Features

- Compare with model based on λ1se, the one with λmin shows:
    1. Lower sensitivity since less features are selected.
    2. Higher specificity since less features are selected.
- As number of features increases, the gap between Recalls on λmin and λ1se shows increase on both sensitivity and specificity.
- The reason why higher number of features leads to higher gap,, may because that it leads a model with more correlations, which makes it harder to be captured by the model.

Part 2 –
Selecting features with confidence

Step 1

Use F statistic to select 200 features

Step 2

Implement LogisticRegressionCV with one vs rest on the dataset and encode the weights into binary codes.

Step 3
Gaining confidence in the selection

3.1 Bootstrapping Simulation

- Here we still use the LogisticRegressionCV as above to find relative values as above.
- We introduced bootstrap 100 times, each time use 35% of the data.

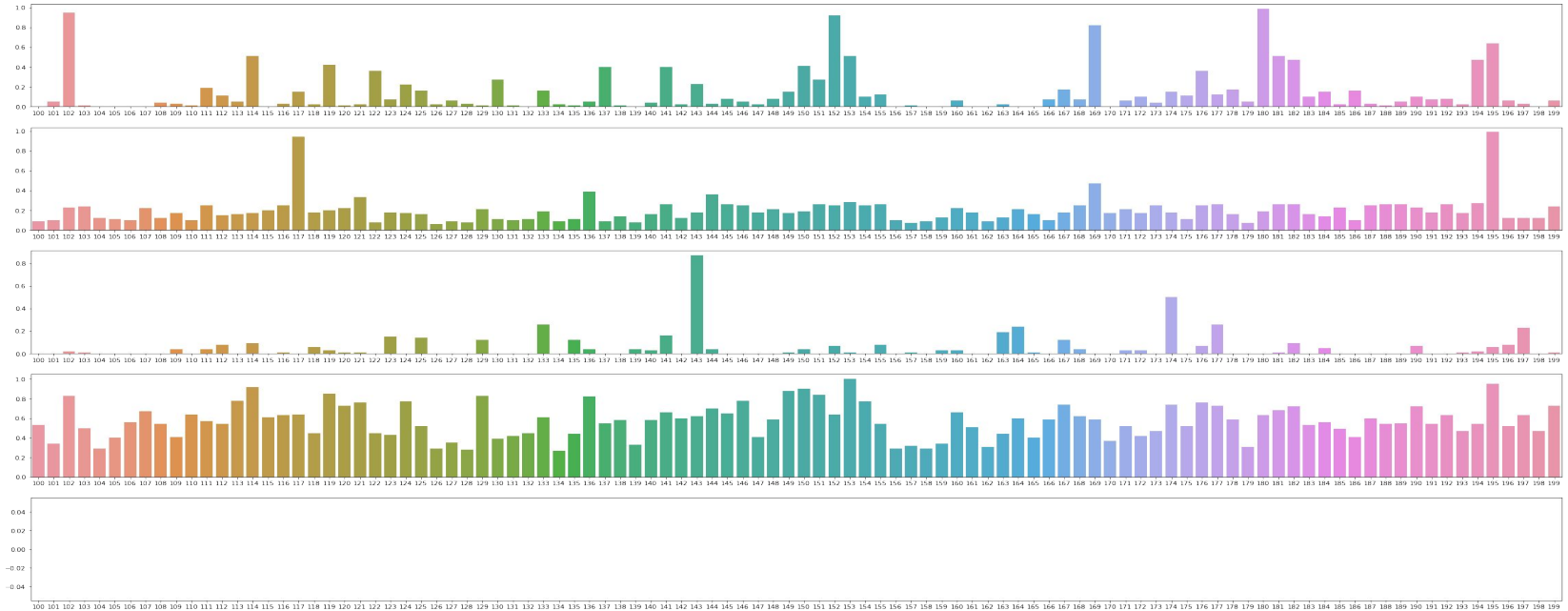
Step 3
Gaining confidence in the selection

3.2 Result Visualization

- We draw the histplot for the feature selected frequency of each class based on the 100 times simulation.
- For better visualization, we draw the histplot for first half and second half of features separately.

Step 3
Gaining confidence in the selection
Feature 0-99

Step 3
Gaining confidence in the selection
Feature 100-99



Step 3
Gaining confidence in the selection

3.3 Top 5 Important Features

- Here we treat the frequency from above on each class as the importance for each feature.
- The top 5 important features and their frequencies are shown on the right.
- For the last class, as COAD, there are only top 4 important features, since frequencies for all other features are zero.

```
Class : PRAD
Feature : [ 86 180  75 152  98]
Importance : [1.   0.99 0.99 0.95 0.94]

Class : LUAD
Feature : [  3 195  65  58  39]
Importance : [1.   1.   0.99 0.97 0.96]

Class : BRCA
Feature : [ 40  69  33   6 143]
Importance : [1.   0.99 0.95 0.93 0.9 ]

Class : KIRC
Feature : [153 195  75 150 114]
Importance : [1.   0.97 0.96 0.95 0.95]

Class : COAD
Feature : [98 13 74 97 72]
Importance : [1.   1.   1.   0.01 0.  ]
```

Thanks for Reading !