Project 3: Feature Selection

Pramod Kumar

March 6 (Spring break), 2020

Abstract

Having large huge number of features doesn't guarantee the performance of the model unless feature actually influence the target class. Feature selection has become really important research problem as recent data-set contain hundreds of thousands of features, for example Image and videos has pixel as a feature if a image is 1920x1080 processing/Learning time will increase exponentially. Hence feature selection plays a crucial role in increasing the systems performance.

Contents

1 Introduction

One famous saying in machine learning is "Noise/garbage goes in garbage comes out". if we feed feature which have good relation with target we are expected to get good results. apart from this we have following advantages [2].

- It enables the machine learning algorithm to train faster.
- It reduces the complexity of a model and makes it easier to interpret.
- It improves the accuracy of a model if the right subset is chosen.
- It reduces over-fitting.

1.1 Filter method

Filter method doesnt depend on machine learning algorithm as we dont train over model, instead we trying to evaluate each feature based on statistical test and there correlation with target class. depending upon input and output type(continues/discrete) we can use Person correlation, LDA, Avona, Chi-square, VR/AVR. In this project I am using VR, which can be defined as [1]:

$$VR(F) = \frac{Var(S_F)}{\frac{1}{C}\sum_{k=1,..,c} Var_k(S_F)} \tag{1}$$

Where F is particular feature, c is class, Var(S_F) is Cross-class variance for Feature F

Basic idea is calculate subset of feature based on cross class variance and train the model and evaluate the performance.

Figure 1: Overall Filter method

1.2 Wrapper method

In wrapper method we take subset of features, train the model and compare the results. These methods are usually computationally very expensive. Based on initial set and add/delete we can define 3 type of wrapper methods

- Forward selection: where initial set is empty and we can keep adding feature one by one till we hit the stopping condition.

- Backward selection: Start with all the features and keep removing one by one till we hit the stopping condition.

- Recursive feature elimination: It is a greedy approach, we create different model and find out best performing features in each iteration. rank the features and select top performers.

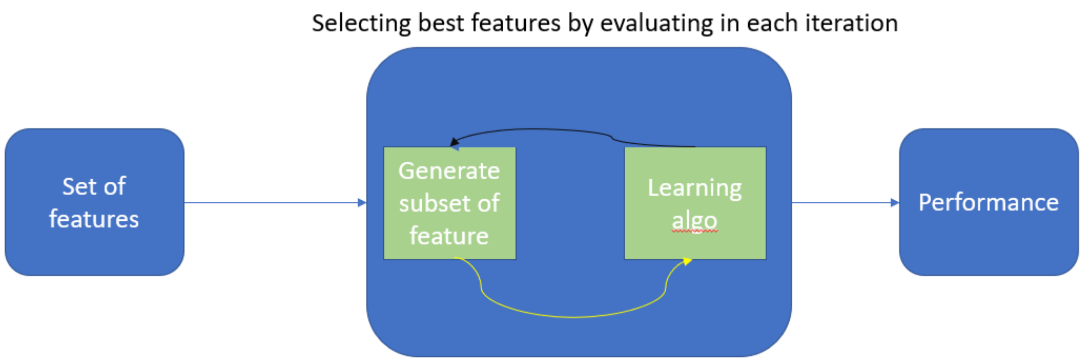

Figure 2: Basic Wrapper method

Since Wrapper method is computational expensive when number of features become large. So similar to Recursive feature elimination, we will implement a intermediate method which will use based on filter method to reduce the feature space and wrapper methods to find the best features.

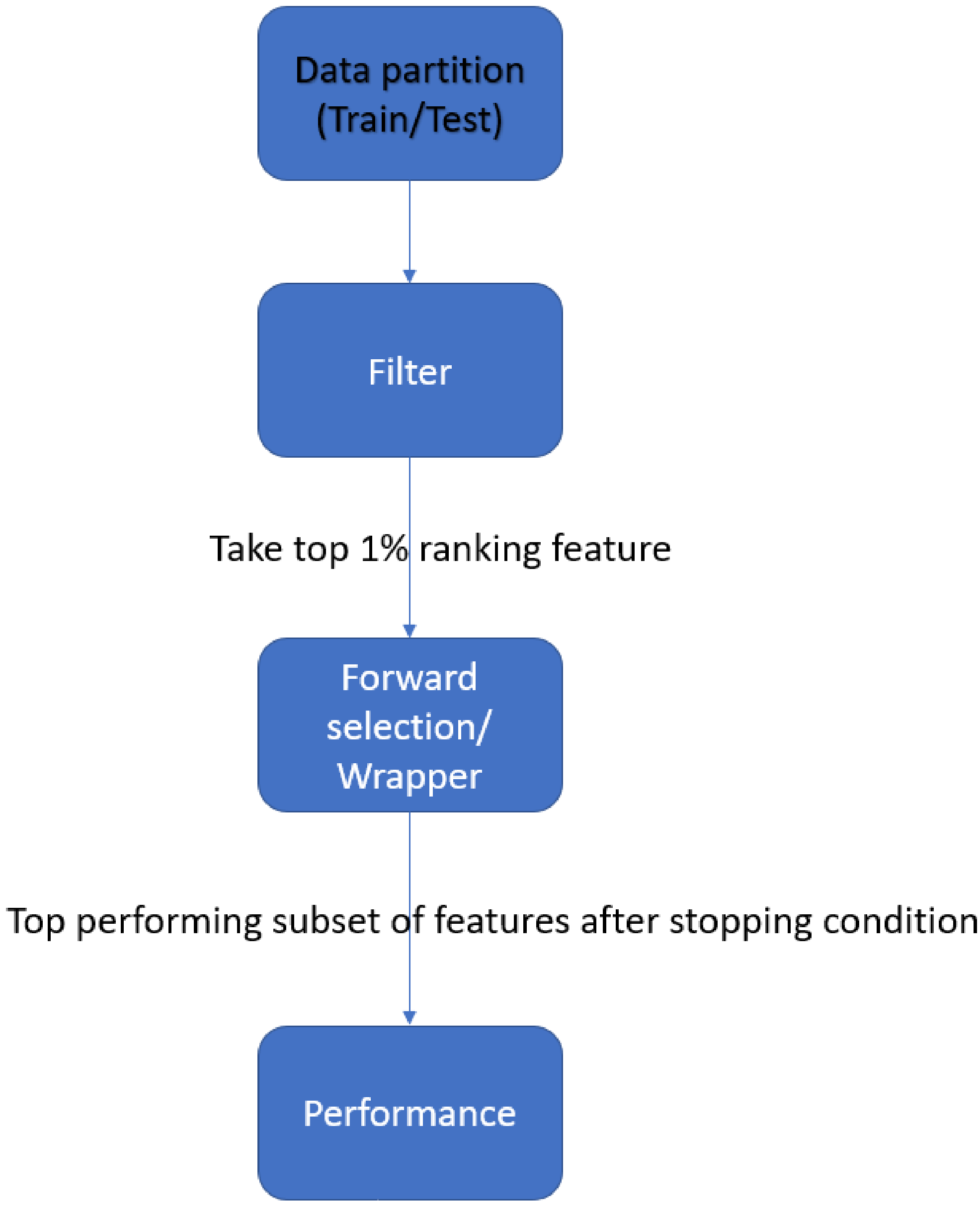

Figure 3: Our hybrid implementation

Over all implementation is as follows:

- Find correlation (with target class) features using filter method i.e using VR, sort them and separate top 1% features.
- Now iterate over those top 1per features, one by one and evaluate using FDA from project 2 and based on performance add them in selected features set. Used error rate for stopping condition.
- Repeat this process 10 to 100 times and calculate average of confusion matrix, accuracy.

2 Results

2.1 Face dataset (over 100 iteration)

[4] Results with 100 Iteration:

Filter step, shows the most discriminating feature between male and female class are located around nose, eyebrows and chin.

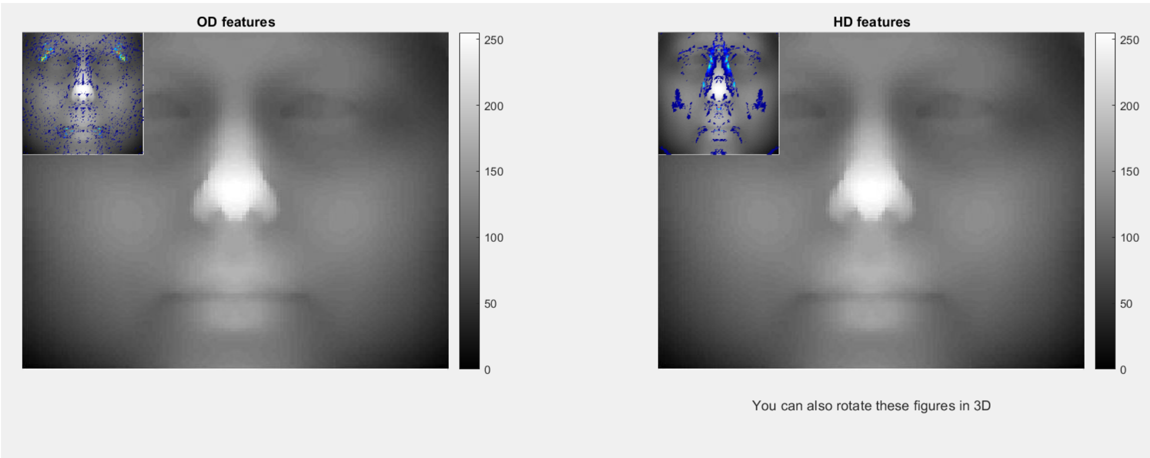

Figure 4: Top 20 feature in top 1% at filter step

After using forward selection method, we can see that subset of feature(from filter step) has equal descriptive power.

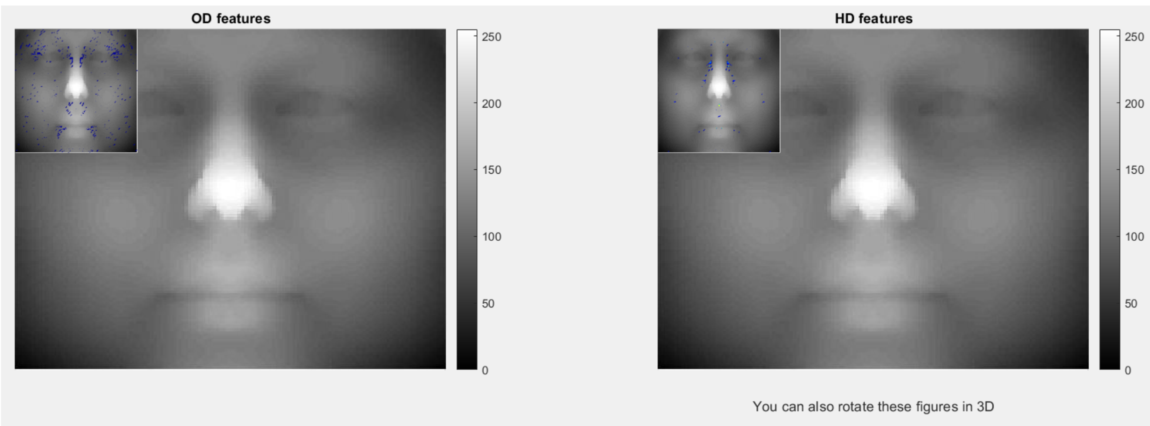

Figure 5: Features after wrapper method(forward selection approach)

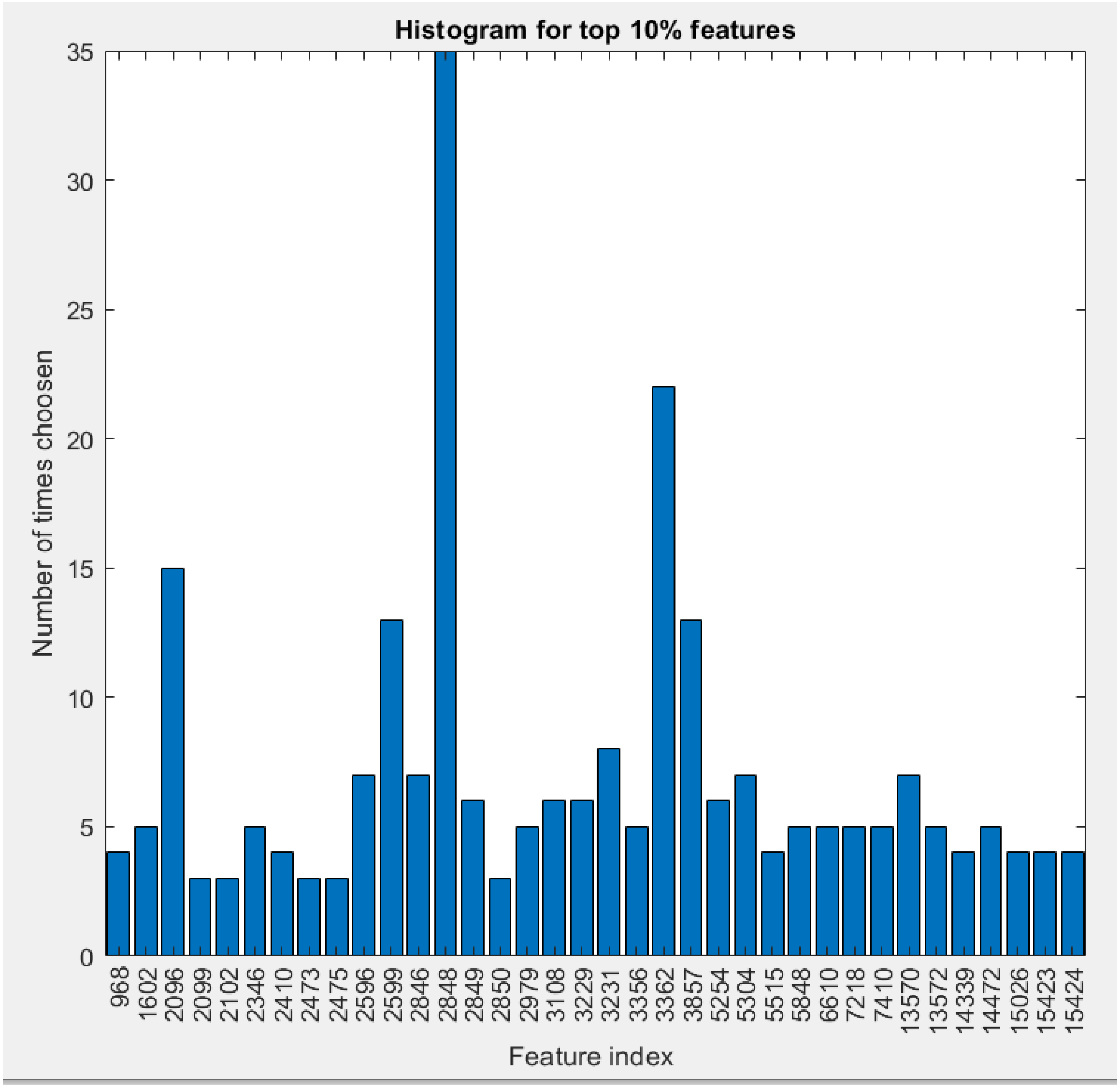

Figure 6: Most selected feature over 100 iteration

Above figure shows top 10% features selected over 100 iteration.

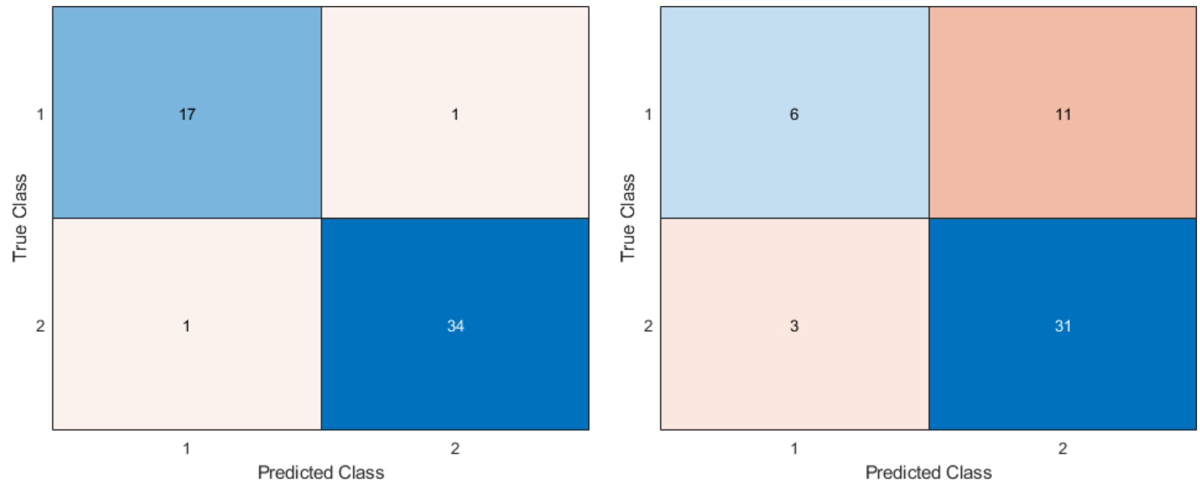

Figure 7: Confusion matrix for training and testing data

```
avg_train_ClassMat =

    0.9489    0.0511
    0.0149    0.9851
```

```
avg_test_ClassMat =

    0.3482    0.6518
    0.0921    0.9079
```

Figure 8: Classification matrix for training and testing data

Iteration	Train acc	Train SD	Test acc	Test SD
10	94.29%	0.0606	60.74%	0.4347
50	95.90%	0.0427	60.38%	0.4347
100	96.70%	0.0314	62.81%	0.3962

Table 1: Average accuracy and Standard deviation

Average accuracy is increasing with increasing number of iterations and SD is improving in same fashion.

2.2 EEG dataset (over 100 iteration)

Following the implementation defined in section 1,

Below Two Figure shows top 20 features in top 1% feature selected with VR, and score for each electrode in filter method.

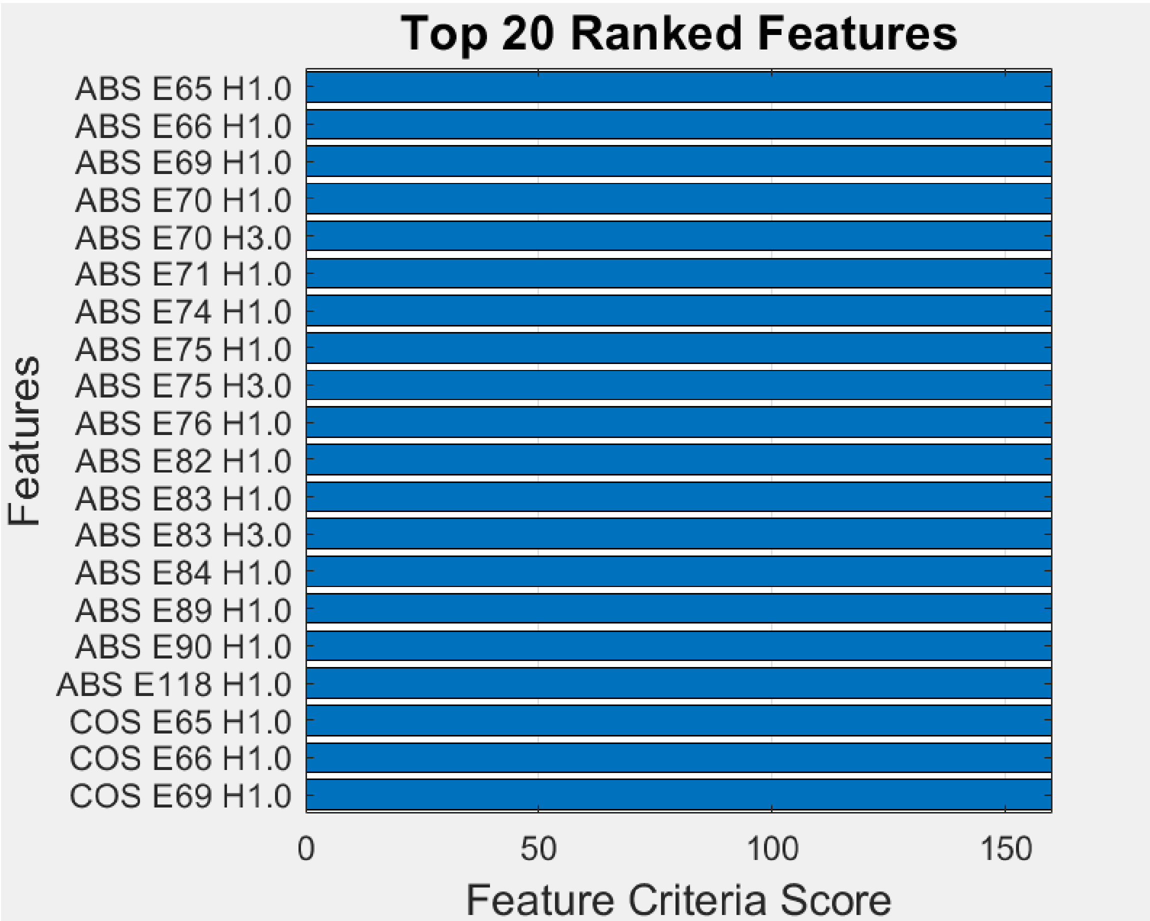

Figure 9: Top 20 feature in top 1% at filter step

Above are the top 20 (out of top 1%)features from filter step, all looks to be of equally important, we will refine them in Wrapper method.

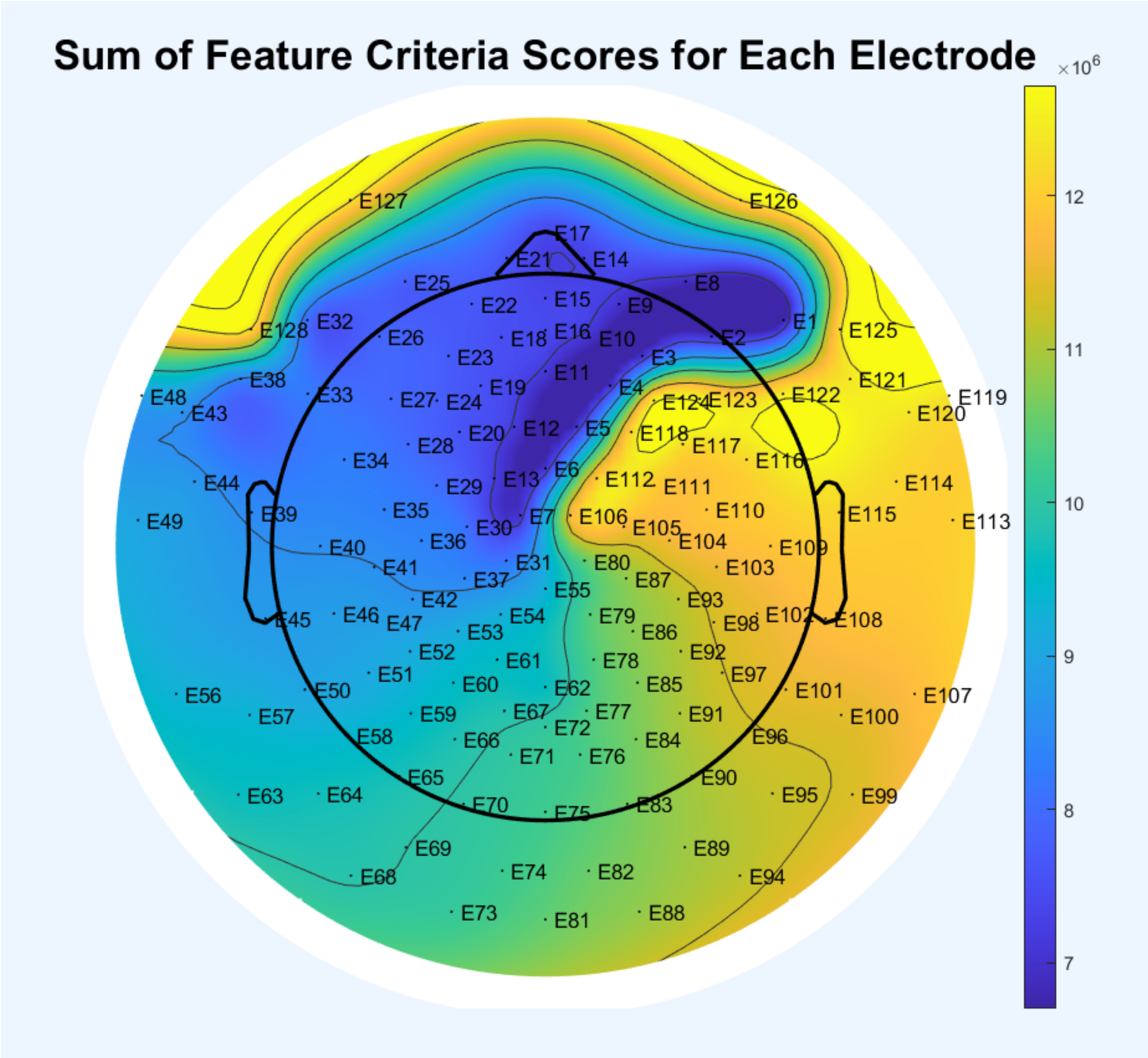

Figure 10: Sum of feature criteria scores for each electrode in filter method

After selecting top 1% features from filter method output, we started with empty set and started adding features 1-by-1. decision of keeping or reject the features is based on Fisher criteria to evaluate discriminate power of each feature. Below Two Figure shows top 20 features in top 1% feature selected with VR, and score for each electrode in Wrapper method.

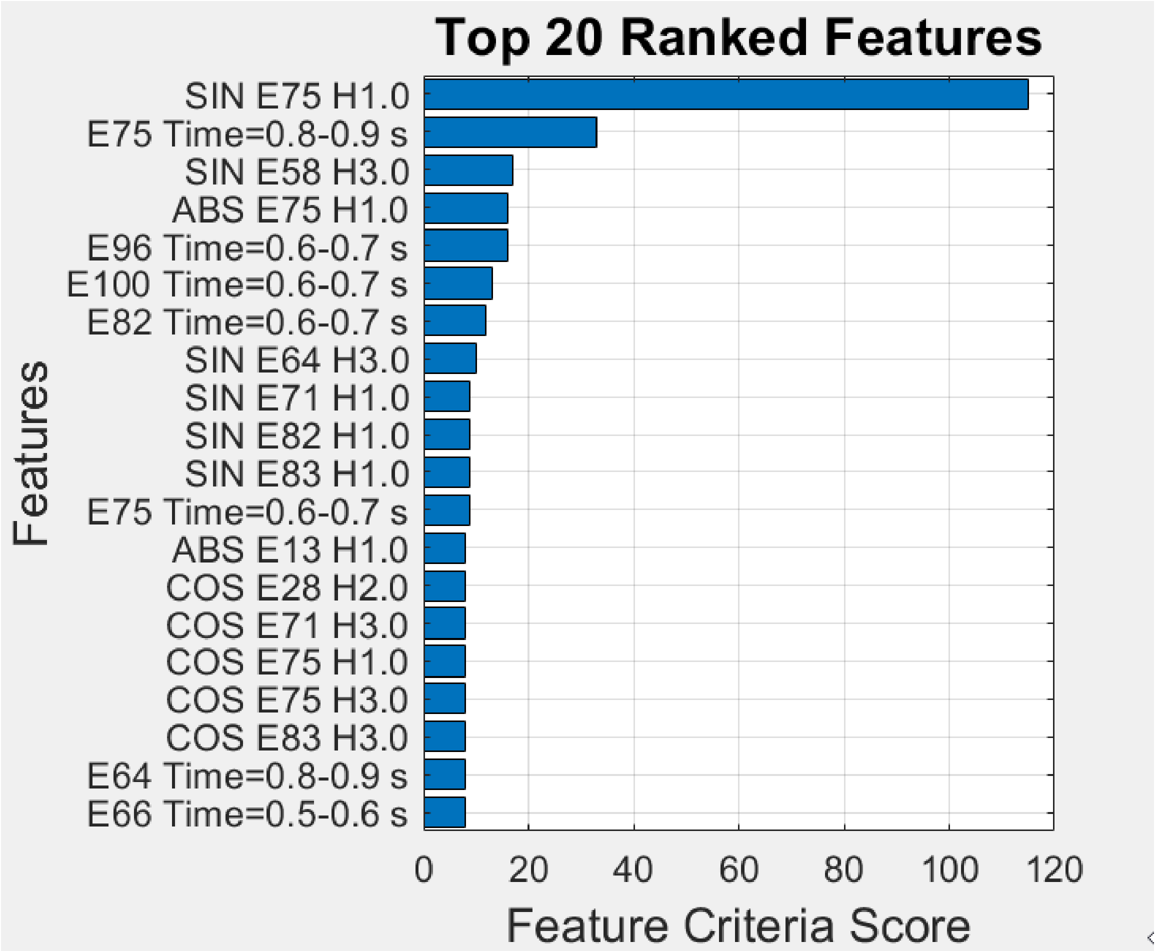

Figure 11: Top 20 feature in top 1% at wrapper step

Above are the top 20 (out of top 1%)features from wrapper step, Over all the iteration SIN E75 and E75 was selected the most.

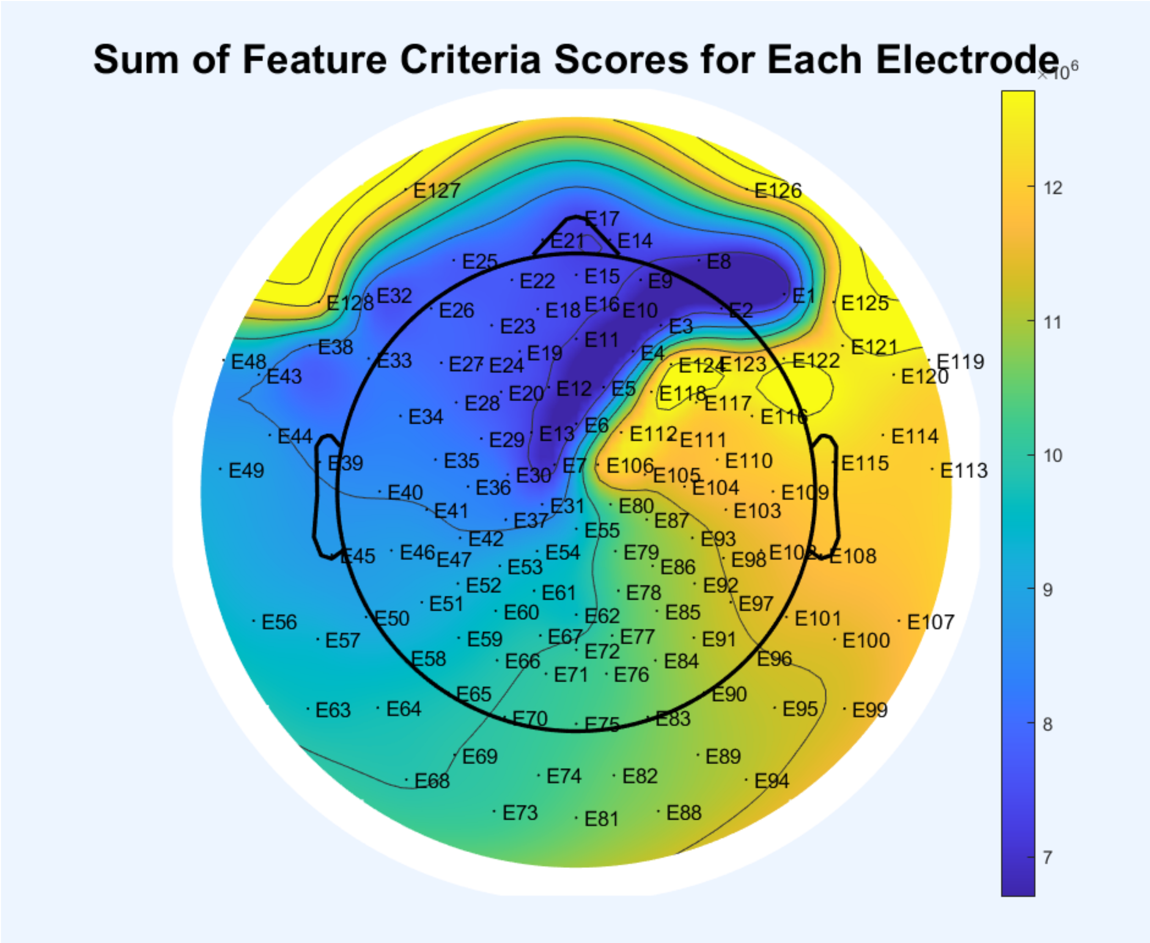

Figure 12: Sum of feature criteria scores for each electrode in Wrapper method

Results with 100 Iteration:

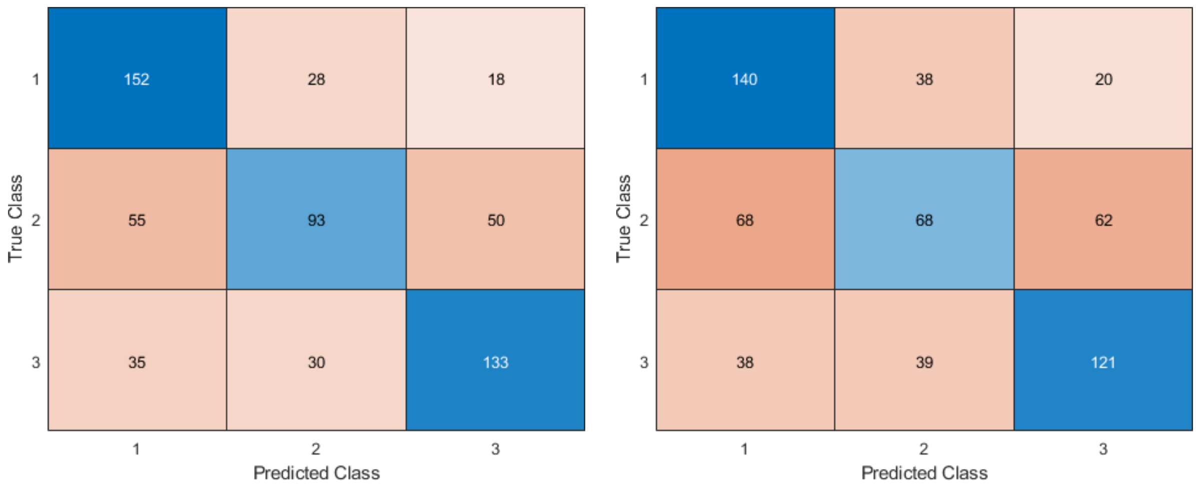

Figure 13: Confusion matrix for training and testing data

```
avg_train_ClassMat =

    0.7698    0.1418    0.0884
    0.2792    0.4697    0.2511
    0.1768    0.1492    0.6740
```

```
avg_test_ClassMat =

    0.7094    0.1907    0.0999
    0.3433    0.3433    0.3134
    0.1918    0.1994    0.6088
```

Figure 14: Classification matrix for training and testing data

The table 3 Compare accuracy and Standard deviation from different number of iteration

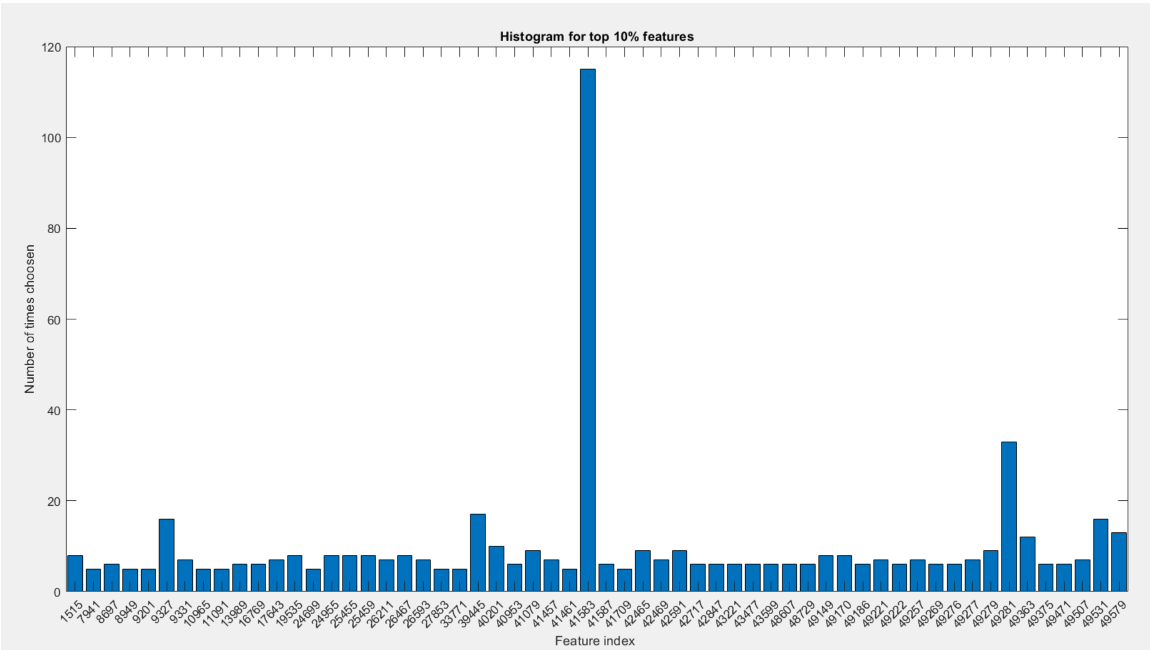

Figure 15: Most selected feature over 100 iteration

Iteration	Train acc	Train SD	Test acc	Test SD
10	63.72%	0.1541	54.93%	0.1967
50	63.86%	0.1576	55.29%	0.1943
100	64.78%	0.1549	55.38%	0.1905

Table 2: Average accuracy and Standard deviation

In above table, it shows that by increasing number of iteration we can we are able to improve average accuracy.

3 Foot dataset Results

Procedure: First convert 4D data into 2D data, append the labels (0 for male and 1 for female). Then shuffle the data and divide into training and test sets. After data is divided follow same procedure from Section 1 & 2.

Results

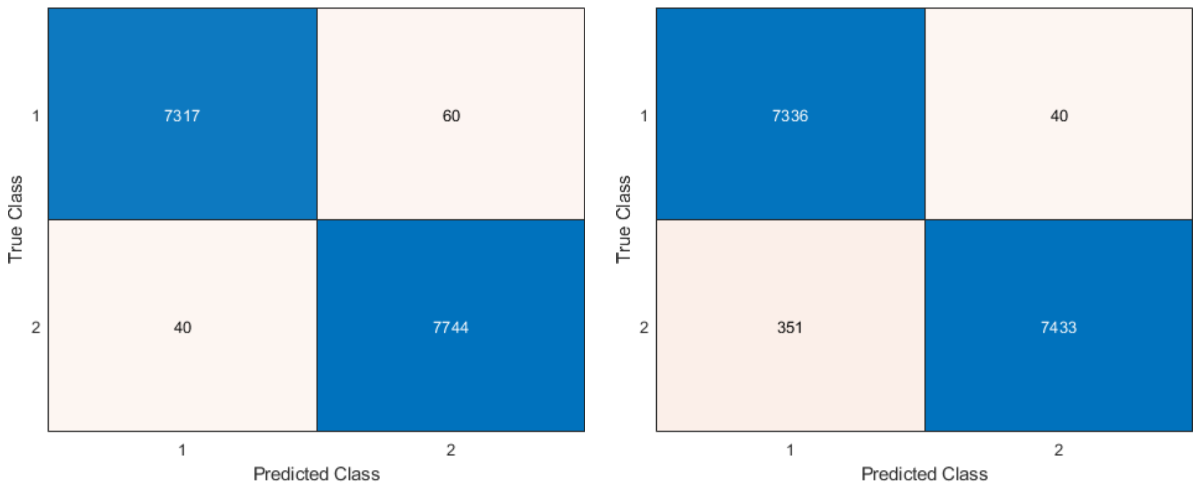

Figure 16: Confusion matrix for training and testing data

```
avg_train_ClassMat =

    0.9919    0.0081
    0.0051    0.9949
```

```
avg_test_ClassMat =

    0.9946    0.0054
    0.0451    0.9549
```

Figure 17: Classification matrix for training and testing data

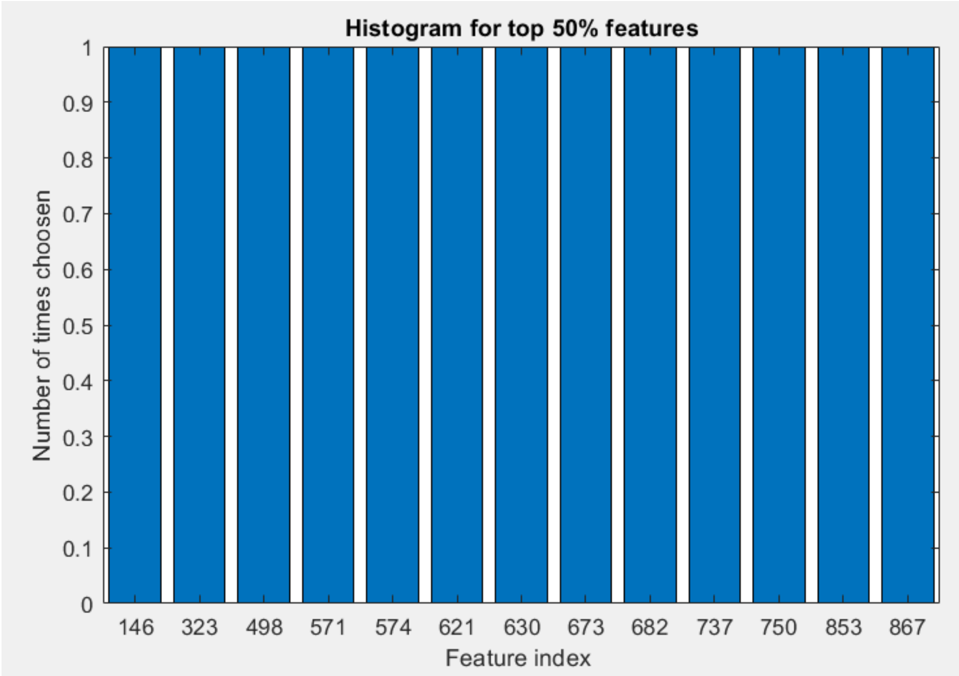

Figure 18: Top 50% feature after Wrapper method

Iteration	Train acc	Train SD	Test acc	Test SD
1	0.9934%	0.0021	0.9747%	0.0281

Table 3: Average accuracy and Standard deviation

3.1 Conclusion

Pure wrapper method is computationally expensive and pure filter method doesn't evaluate method hence suffer from lots of redundant features. By extracting best of both approaches we can get the acceptable result with less computation. This implementation takes 4 minutes for Face dataset and 20 minutes for EEG dataset for 10 iteration on i7 processor.

References

[1] Bishop, Christopher M. Pattern Recognition and Machine Learning. 2

[2] Saurav Kaushik, Introduction to Feature Selection methods 2

[3] Causal Feature Selection, I. Guyon et al.(to appear in "Computational Methods of Feature Selection", Liu-Motoda Eds. Chapman&Hall, 2007

[4] A Quantified Study of Facial Asymmetry in 3DFaces, Yanxi Liu Je Palmer