Feature Selection

Skanda Bharadwaj

June 13, 2019

Feature selection is the process of selecting a subset of relevant features for use in model construction. Feature selection techniques are used for: simplification of models to make them easier to interpret by researchers, shorter training times, to avoid the curse of dimensionality, and enhanced generalization by reducing overfitting. This project addressess two major feature selection methods – Filter and Wrapper. While the filter method ranks the features based on some discriminative power, wrappers use a search algorithm to search through the space of possible features and evaluate each subset by running a model on the subset.

Contents

1 Introduction

Feature selection is one of the most important steps in Machine Learning, which not only decreases the computational complexity by reducing the dimension of the feature space, but also removes the redundancy in the feature space by removing the features that are not contributing towards classification. In this project, we explore two techniques of feature selection −

- Filter Methods
- Wrapper Methods

Filter methods are basically preprocessing steps. In filter methods, a discriminative power is used to rank the features to choose best features among the available ones in the feature space. On the other hand, in wrapper methods we try to use a subset of features and train a model using them. Based on the inferences that we draw from the previous model, we decide to add or remove features from your subset. In this project we incorporate a combination of both the selection methods called *"filtrapper method"*. We first rank the features based on some discriminative power and then use one of the wrapper methods to create the subset of features using which we train and test the model.

2 Feature Selection Methods

In this section, we describe the two feature selection methods used in the project.

2.1 Filter Methods

Filter methods are basically preprocessing steps. In filter methods, a discriminative power is used to rank the features to choose best features among the available ones in the feature space. The selection of features is independent of any machine learning algorithms. Instead, features are selected on the basis of their scores/ranks in various statistical tests for their correlation with the outcome variable. The selection criteria can be based on statistical measures such as *variance ratio* (VR) or *augmented variance ratio*(AVR). The selection criteria VR and AVR are defined as −

- **Variance Ratio (VR):** For a feature F with values S_F in a data set with C total classes, the *variance ratio* (VR) of between- to within-variance is calculated as

$$VR(F) = \frac{Var(S_F)}{1/C \sum_{k=1}^{C} Var_k(S_F)} \tag{1}$$

  where, $Var_k(S_F)$ is the variance of the subset of values from feature F which belongs to class c. Though VR accounts for the spread between means with the between-class variance, it does not take into account separation of classes on an individual basis.

- **Augmented Variance Ratio (AVR):** It is possible for a feature to have small within-class scatter for a class even if the mean value of the feature for that class is very close to the mean value of the feature in another class. Thus *augmented variance ratio* (AVR) is defined as −

$$AVR(F) = \frac{Var(S_F)}{\frac{1}{C}\sum_{k=1}^{C} \frac{Var_k(S_F)}{min_{i \neq j}(|mean_i(S_F) - mean_j(S_F)|)}} \tag{2}$$

where $mean_i(S_F)$ is the mean of feature Fs values in class i. AVR is the ratio of the variance of the feature between subjects to the variance of the feature within subjects, with an added penalty for features that have close inter-subject mean values. Individual features that have higher variance ratios are more discriminative.

Figure 1 represents a schematic of the Filter method

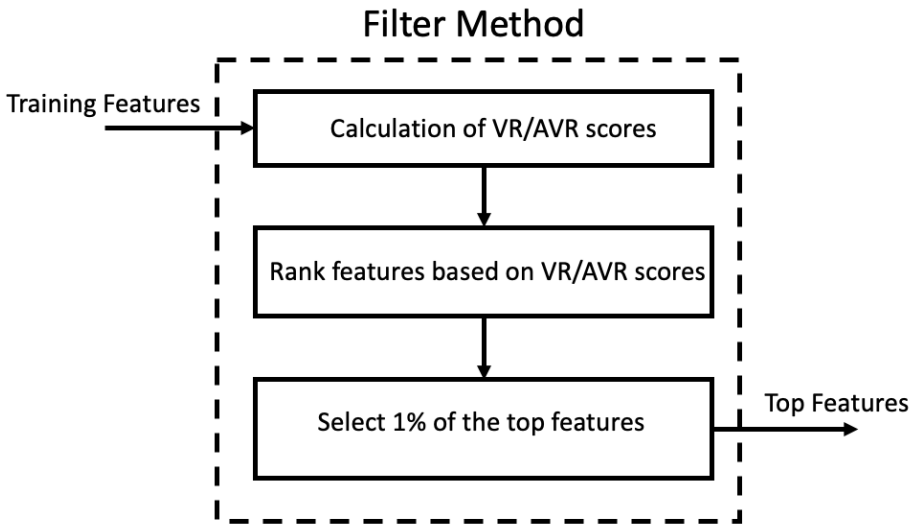


Figure 1: Schematic of Filter Method

2.2 Wrapper Method

In wrapper methods, a subset of features is used to train a model. Based on the inferences from the previous model, features will either be removed or added to the subset. The problem is essentially reduced to a search problem. These methods are usually computationally very expensive. Some common examples of wrapper methods are forward feature selection, backward feature elimination, recursive feature elimination, etc. In this project, Sequential Forward Selection is used to create the subset of features.

- **Sequential Forward Selection:** Sequential Forward selection is an iterative method in which there are no features in the model initially. In each iteration,

the feature which best improves the model is added till an addition of a new feature does not improve the performance of the model. In this project, maximizing the overall accuracy is used as the stopping criterion. If accuracy doesn't increase for more than 10 iterations, the algorithm will be terminated.

Figure 2 represents the schematic of Sequential Forward Selection.

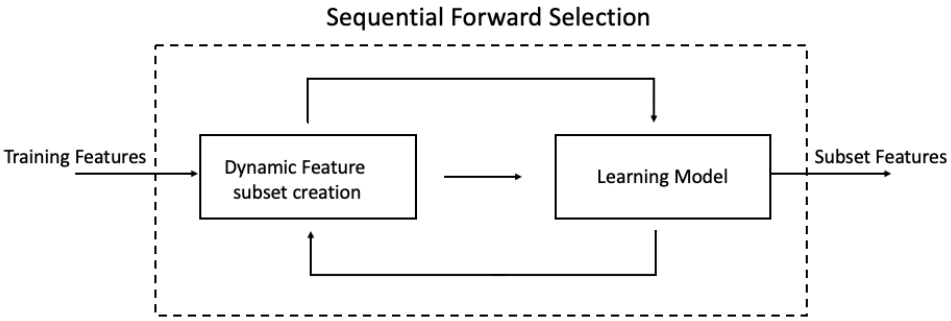


Figure 2: Schematic of Sequential Forward Selection

2.3 Filtrapper Method

It is a well-known fact that the wrapper methods are computationally very expensive and therefore, to reduce the complexity, wrapper based search is done on the output of the filter. Filter produces an output that is only 1% of the total available features in feature space and wrapper methods are carried on these 1% top-features. Figure 3 shows the filtrapper method.

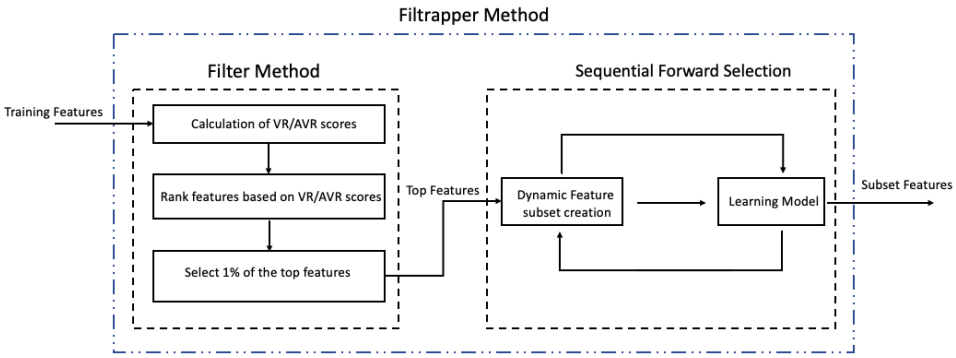


Figure 3: Schematic of Filtrapper Method

3 Data

In this section we discuss the data we are interested in detail.

3.1 Dataset

The dataset is a digitized scan of different subjects with neutral facial expression. Each face has 2 kinds of features called *OD* and *HD* features. *OD* and *HD* stand for *Orientation Difference* and *Height Difference* respectively. Each face contains a total of 7751 features of both *OD* and *HD* features .

The parameters of the wine data set is as follows -

- Number of Classes : 2
- Number of Features : 15,500
- Number of Observations : 104

4 Procedure and Code Flow

Procedural caveats −

- In the Filter block of the above-mentioned filtrapper algorithm has provision to choose between the 2 criteria − VR and AVR. In the results section, an exhaustive summary has been reported where AVR is used as the selection criterion. Only specific results are reported for VR (for a random trial).
- The entire dataset is split into 70% training data and 30% testing data based on a popular thumb-rule. 1% of the top-features are selected from the filter module using AVR.
- The top-features are sent into the wrapper module for the choice of suitable subset. As mentioned earlier, sequential forward selection is used as the search method and Linear Discriminant Analysis as classifier.
- In the wrapper module, the training data is again split into training data and cross-validation data to perform classification. Again, the splitting is done based on the popular thumb-rule − 70-30. To reduce the redundant increase in the complexity, maximization of accuracy on cross-validation set is used as the stopping criterion. If the addition of features into the subset does not result in the increase of the accuracy for over 10 iterations, the algorithm is terminated.
- The classification of the testing data is done using the feature subset from the wrapper module. To substantiate the results, the entire procedure is run a 100 times and the results are averaged.

Figure 4 represents the flow-chart of the filtrapper code on a bigger picture.

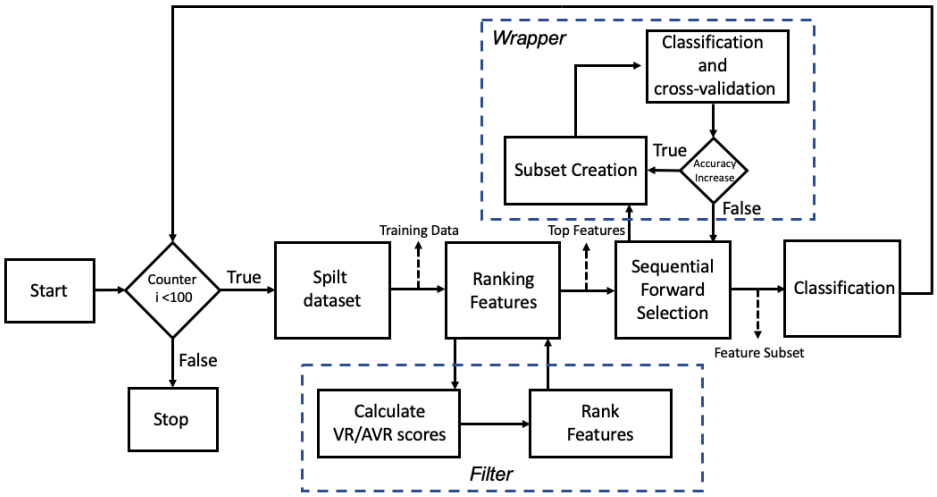
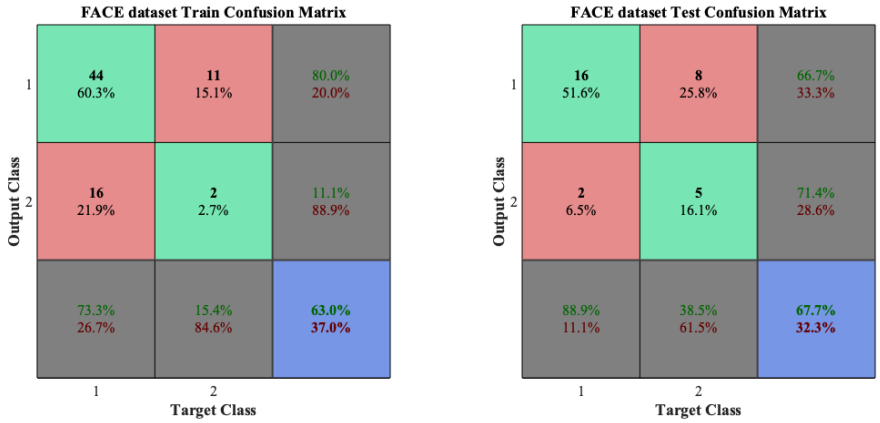


Figure 4: Flow-chart of Filtrapper Method

5 Results and Discussions

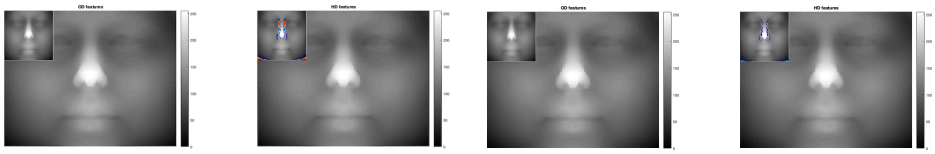
Given below is the confusion matrix that summarizes the results of the filtrapper method for one of the iterations. From a randomly selected feature subset of the 100 iterations, classification was carried out on both training and testing data. Figure 5 represents the results,



a b

Figure 5: Figures *a* and *b* represent the confusion matrices of training and testing of the FACE data set.

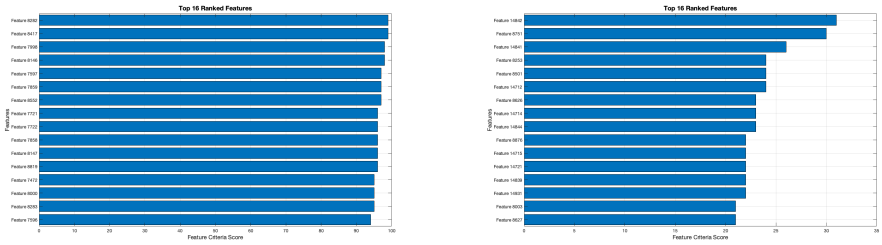
Figure 6 represents the visualization of top 1% most discriminative features.



a b

Figure 6: Figures *a* represents the the number of times a feature has been in the top 1% of the features and Figure *b* represents the number of times a feature that has been selected to the feature subset by SFS.

Figure 7 represents the visualization of the histogram of the top 10% features selected via forward selection



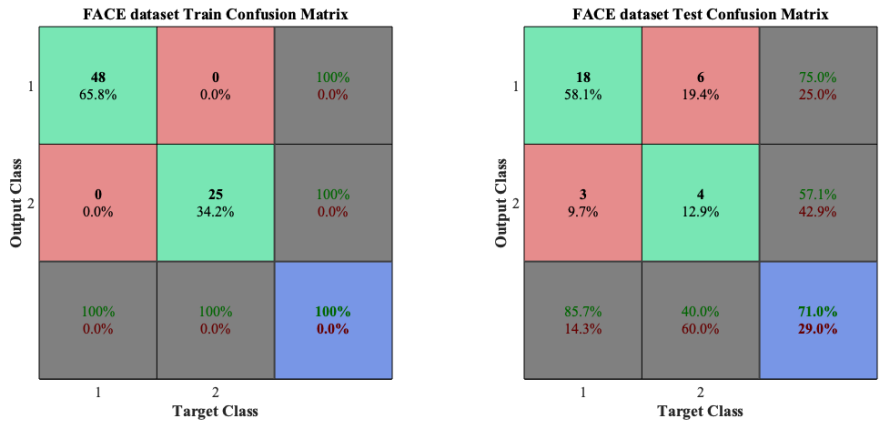
a b

Figure 7: Figure *a* represents the top 10% of the features selected with the number of times they have been within the top 10%, and figure *b* represents the values of the selected features.

***The accuracy averaged over a* 100 *iterations for the FACE dataset with AVR scoring and SFS was found to be* 65.26%. *The maximum and minimum accuracies were found to be* 83.87% *and* 45.16% *respectively with a standard deviation of* 9.18%.**

6 Conclusion

For the FACE dataset, the classifier was run using all the $15,500$ features. The confusion matrix is shown in Figure 13.



a b

Figure 8: Figures *a* and *b* represent the confusion matrices of training and testing of the FACE dataset with all 15, 500 features.

It can be seen that, although the training accuracy is 100%, the testing accuracy is as less as 71%. This indicates overfitting. Very large number of features not contributing towards classification can be often considered as noise and can lead to deteriorated performance. But, with feature selection, the number of features reduced drastically. It can be seen that the training accuracy and testing accuracy are reasonably comparable and is thus a better generalization of the dataset. Not only did the feature selection maintain a reasonable accuracy, but also reduced the dimensionality of the feature space there by reducing the computational complexity by a very huge factor. It was also observed that the classifier couldn't even be run for the EEG dataset with 49,920 features in MatLab. Therefore, feature selection is invariably a very important step for large-scale machine learning.

References

[1] Liu, Yanxi and Jeff Palmer. *"A Quantified Study of Facial Asymmetry in 3D Faces."* AMFG (2003).

[2] Yanxi Liu, Karen L Schmidt, Jeffrey F Cohn, Sinjini Mitra, *Facial asymmetry quantification for expression invariant human identification*, Computer Vision and Image Understanding, Volume 91, Issues 1–2, 2003, Pages 138-159, ISSN 1077-3142,

[3] https://en.wikipedia.org/wiki/Feature_selection

[4] https://www.analyticsvidhya.com/blog/2016/12/introduction-to-feature-selection-methods-w