A close up of a sign

Description generated with high confidence

# Feature Selection Algorithm

|  |  |
| --- | --- |
| Version | 1.0 |
| Document Title | Wrapper Description |
| Group Members | Tanuj Gyan, Sayanti Ray, Raviprakash Shreeraksha, Babak |
| Date of Submission | 26-March-2018 |
| Instructor Name | Professor Johnson P Thomas |
| TA Name | Vasanthi |

## Wrapper Method Description:

The wrapper method assigned to our group is Brute force, Forward and Backward. This means that we will have to look at all possible datasets created by including and excluding the features and run the machine learning algorithm to check which subset gives the most accurate results. In our data set we have 760 features and there should be one data set for every combination of the feature so we will have two possibilities for every dataset – include it or exclude it. This will lead to 2^760 different possibilities which is cumbersome to calculate. Lets say for example we start our data set with all 760 features and start dropping one feature at a time. So next iteration will run with 759 features and there are 759 different datasets with 759 features and so on. The computing power needed in this case will be high. We have to write our program in such a way that this dropping of column and creation of data set happens automatically and the accuracy is stored for every data set. The accuracy has to be compared to threshold (0.03) and data sets that fall below this accuracy level will be discarded. At the end we have to choose the most ideal candidate and with least number of features.

In “forward selection method”, which is an iterative method, we begin with having no feature in the model. In each iteration, we continue adding the feature which best enhances our model till an addition of a new variable does not enhance the performance of the model. In other words, we start by measuring the error of the one-component subsets, Y1, Y2, ..., Y760; so that we can find the best individual feature, let’s say y(1). Next, we find the best subset comprising of two components or features, y(1) and one other feature from the remaining 759 input features. Let’s assume y(2) is the other attribute in the best pair alongside y(1). Subsequently, the input subsets with three, four, and more features are evaluated. Finally, the best subset with n features is the m-tuple comprising of y(1), y(2), ..., y(n), while overall the best feature set is the winner out of all the 760 steps.

Alternatively, in “backward elimination method”, we begin with all the features and removes the least significant feature at each iteration that enhances the performance of the model. We repeat this until no improvement/change is observed on removal of features.

To kickstart the process we did some manual experiments with KNN to see how accurate the modal will be if we use all 760 features. We did the experiment with

|  |  |  |
| --- | --- | --- |
| K | Total labels | Matching labels |
| 1 | 11602 | 2133 |
| 9 | 11602 | 2313 |
| 21 | 11602 | 2500 |

Point to be noticed was as we were increasing the number of neighbors the accuracy was increasing. However the project limits us to use up to K=9, we did it with K=21 for the purpose of seeing the change in accuracy that can be achieved by increasing the number of neighbors.

We also tried removing the first feature from test data and train data and for K=1 the matching labels dropped to 531 which is approximately 1/4th of the number we achieved with 760 features. This difference might be there because of first feature being more significant in data prediction process. However we will be automating the process of dropping features and calculating accuracy in subsequent days to achieve better understanding of data.