AI and Machine Learning Solutions to Advanced Business Analytics, LIVE ONLINE
Instructor-led Live Online Training, Starting Jan 25th
Sign Up Now | Use Code: Scale2020 | Save $1,000!
ENROLL NOW

Fighting Cancer through Data, AI & IoT
Fighting Fraud with Machine Learning
Fighting Terrorism with Data Science
JOIN US FOR A DATA SCIENCE FOR SOCIAL GOOD HACKATHON
DATA SCIENCE FOR SOCIAL GOOD
GET PAID TO EARN YOUR MSDA
Apply Now!

DATA SCIENCE CENTRAL

A Complete Guide to Sequential Feature Selection

POSTED BY

Tarek Dib
Machine Learning Engineer, Data Scientist, Software Developer, Mathematics Enthusiast

Comments | 0 Like | 0 Tweet | 0 Share | 0 Shares | 0 Views

Comments: 0 | Views: 1,7K

Comments: 0 | Views: 1,7K

Comparing features, feature selection is the procedure of choosing a subset of the features from the data so that the output of the model can be accurate and according to the requirements. Sometimes, the features in the data are more than what is needed for various factors such as the limitations of the time that there are various features given in the data which are not even required for the generation of the models and also the presence of these features can affect the performance level of the model. In modelling, it becomes important to select the features.

Basically, the feature selection is a procedure where we choose from the dataset so that the model can perform better and the computational efforts will be reduced. In feature selection, we try to find out major variables from the set of major variables which are possessing a strong relationship with the output variable. This means changes in an input variable should have changes in the output variable.

Agenda

There are various reasons to perform feature selection.

- Simplification of the model.
- Less computational time.
- To avoid the curse of dimensionality.
- Improve the compatibility of the model with data.

Roughly the feature selection techniques can be divided into three types of methods.

- Filter method
- Wrapper method
- Embedded method.

Filter methods

These methods are very fast and easy to do the feature selection. In this method, we perform feature selection at the time of preprocessing of the data. Generally, these methods are based on the correlation criteria. Algorithm in this process, first, it performs some statistical test so that it does not remove the multicollinearity from the data. The have performance of the modelling are not affected and then method of feature selection is as follows.

Some of the techniques under the filter methods are

- Information gain
- Chi square test
- Fisher's score
- Correlation coefficient.
- Variance threshold
- Mean absolute deviation.
- dispersion coefficient

Wrapper Methods(Greedy Algorithms)

In this method, feature selection algorithm is to wrap the model with a wished number of subsets of features in an iterative way. In this method, the feature selection is a search problem where various subsets and it iterates the number of feature pairs reduced or increased. The stopping criteria of the iteration are to select the features which will improve the model in modelling. As at the end of the procedure of wrapper method, the optimal set of features gets selected for the modelling.

The basic architecture of the technique followed by wrapper methods is as follows.

Some of the techniques under these methods are:

- Forward selection.
- Backward selection.
- Bi -directional selection.
- Exhaustive selection.
- Recursive selection.

Note: the main motive of the article is to talk about the sequential feature selection which is a technique related to the wrapper methods so here in the article we will discuss the sequential feature selection.

Embedded methods

This method is the combination of the filter method and wrapper methods which is that in this method we use and accounts of the wrapper method. These methods are blended as part of the learning algorithm. This process minimizes the computation time and duration of filter and wrapper methods.

Some of the techniques under this method are:

- Regularization: Ridge regularization methods are
- Lasso regularization.
- Elastic net regularization.
- Ridge regularization.
- Tree based methods. Here tree -based methods are:
  1. LightGBM.
  2. XGBoost.

The basic architecture of any procedure containing embedded methods is as follows feature selection techniques is as follows.

Sequential Feature Selection Algorithms

Sequential feature selection algorithms are a family of greed search algorithms where it adds and removes features from the dataset sequentially. Sometimes it removes and adds features sequentially and selects those features from the dataset which are relevant to the problem. Sequential feature selection is a greedy search algorithm and it is sometimes known as a wrapper for logistic models and the mlxtendlibrary use as a criterion for the classification model.

A sequential search algorithm

This searching algorithm aim at removes the feature candidate from the candidate subset while evaluating the objective function at criterion. Sequential searches follow only one direction either forward or backward and it adds or removes the feature candidate the number of the candidate feature subset.

On the basis of movement, we can divide them into two variants.

Sequential forward selection(SFS)

In SFS variant features are sequentially added to an empty set of features until the addition of extra features does not reduce the criterion.

Mathematically if the input data is the algorithm is

Input: $Y = \{y_1, y_2, ..., y_n\}$

Then the output will be:

Output: $X_k = \{y_j \mid j = 1, 2, ..., k, y_j \in Y\}$ where $k = \{0, 1, 2, ..., d\}$

Where the selected features are a and d is

In the mlxtendlibrary it is used as and d is (where is the size of the subset)

In the mentioning, the size is k. y where y is the number of desired features.

Sequential Backward Selection (SBS)

This variant algorithm picks all the features from the input data and combines them in a set and sequentially removes them from the set until the remnant of further features becomes the criterion.

mathematically if the input data is

$Y = \{y_1, y_2, ..., y_n\}$

The output of the variant will be

Output: $X_k = \{y_j \mid j = 1, 2, ..., k, y_j \in Y\}$ where $k = \{0, 1, 2, ..., d\}$

In the mlxtendlibrary it is a subset of features and is of which is the size of the subset.

In the mentioning, the size is k. y where y is the number of desired features.

There are two more variants of the sequential feature selection.

- Sequential forward floating selection.
- Sequential backward floating selection.

These floating variants are the extensions of the SFS and SBS where they consist of an additional direction of inclusion step or exclusion step after each step of inclusion or exclusion. Generally, these extensions provide a larger number of candidate subsets of features sets.

Let's see how we can implement these all using python.

MLxtend is a package that provides the implementations of sequential feature selection methods.

You can check the whole code in the link.

Here in this article, I will just give the images and will explain to them how did it work in the background.

In this procedure, I am using the iris data set and feature_selection module provided in mlxtend library.

In the following code after defining x, y and the model object we are defining a sequential forward selection object for a KNN model.

from mlxtend.feature_selection import SequentialFeatureSelector as SFS

```
sfs = SFS(knn,
          k_features=(1,4),
          forward=True,
          floating=False,
          verbose=2,
          scoring='accuracy',
          cv=0)
```

sfs.fit(x, y)

Output

```
Features: 1/4 -- score: 0.9666666666666666
Features: 2/4 -- score: 0.9666666666666666
Features: 3/4 -- score: 0.9666666666666666
Features: 4/4 -- score: 0.9666666666666666
```

Here in the output, we can see all the scores with the number of features. Here we have given the range (1,4) in the parameter and our model will fit with all the numbers of the input features in the range (1, 4) inclusive. (You will all the object in below the 10 and floating = False because this is an example of SF but SFS of SBS.

Using sfs.subsets_ we can view check all the results of every step.

```
{1: {'avg_score': 0.9666666666666666,
     'cv_scores': array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1.]),
     'feature_idx': (3),
     'feature_names': ('petal width'),
     'ci_bound': 0.0,
     'std_dev': 0.0,
     'std_err': 0.0},
 2: {'avg_score': 0.9666666666666666,
     'cv_scores': array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1.]),
     'feature_idx': (2, 3),
     'feature_names': ('petal length', 'petal width'),
     'ci_bound': 0.0,
     'std_dev': 0.0,
     'std_err': 0.0},
 3: {'avg_score': 0.9666666666666666,
     'cv_scores': array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1.]),
     'feature_idx': (1, 2, 3),
     'feature_names': ('sepal length', 'petal length', 'petal width'),
     'ci_bound': 0.0,
     'std_dev': 0.0,
     'std_err': 0.0},
 4: {'avg_score': 0.9666666666666666,
     'cv_scores': array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1.]),
     'feature_idx': (0, 1, 2, 3),
     'feature_names': ('sepal width', 'sepal length', 'petal length', 'petal width'),
     'ci_bound': 0.0,
     'std_dev': 0.0,
     'std_err': 0.0}}
```

Here in the results, we can see the results of every step with the name of the feature in the dict.

Just by changing the parameters, we can toggle between all the variants of the sequential feature selection.

For SBS

- forward = False
- floating = False

For SFFS

- forward = True
- floating = True

For SBFS

- forward = False
- floating = True

Changing the results.

```
Features: 3/4 -- score: 0.9666666666666666
Features: 2/4 -- score: 0.9666666666666666
Features: 1/4 -- score: 0.9666666666666666
```

```
Features: 1/4 -- score: 0.9666666666666666
Features: 2/4 -- score: 0.9666666666666666
Features: 3/4 -- score: 0.9666666666666666
Features: 4/4 -- score: 0.9666666666666666
```

```
Features: 3/4 -- score: 0.9666666666666666
Features: 2/4 -- score: 0.9666666666666666
Features: 1/4 -- score: 0.9666666666666666
```

We can also see the results in the DataFrame.

Results of SFS

Results of SBS

Here we can see the SFS and SBS have found three best results in which Sequential feature the procedure of feature selection has different.

In the Conclusion

- $y_{_score}$ = cross validation scores.
- $y_{_std_dev}$ = standard deviation
- $y_{_ci_bound}$ = standard deviation of cross validation scores.
- $y_{_std_err}$ = standard error of the cross validation scores

Visualizing the standard deviation for the SFS

Here in this article, I tried to cover the basics of feature selection and what it becomes so important as part of the modelling process with the data. There are various methods that can be used for feature selection, but the prefer to go with the sequential feature selection because by the time performing the modelling we can have full control of the procedure and we can control it. Sometimes the procedure and filtering methods are faster but they are not accurate. The main motive of the article was to make feature selection automatic but should be in the controlled form. In the reference, I have provided the link for the whole code, so if any reader wants to practice then they can access the code.

- Sequential Feature Selector
- Simple Guide methods for the above video

What Do You Think?

0 Comments

Sort by: Newest

Add a comment...

Newest Comments Plugin

Subscribe to our Newsletter

Get the latest updates and relevant offers by sharing your email.

Email *

By signing, I'm in agreeable to...

Submit

Join our 2Million Group. Be part of an engaging community

THOMAS VERBIN →
Tomas is a graduate in automobile engineering and worked in the industry for 3 years before discovering data science and analytics. He has a strong interest in deep learning and writing blogs on data science and machine learning.

SHARE | COMMENTS | 0 | 0 | 0