Feature Extraction Using WPT/EEMD Documentation

Release 0.1

Melih C. Yesilli, Firas A. Khasawneh, Andreas Otto

Feb 14, 2020

CONTENTS

This toolbox includes the documentation for the Python codes that extract features by using Wavelet Packet Transform (WPT) and Ensemble Empirical Mode Decomposition (EEMD) and diagnose chatter in turning process. Algorithms are based on the methods explained in [1].

CHAPTER

ONE

WAVELET PACKET TRANSFORM (WPT)

1.1 Feature extraction and supervised classification using WPT

This function take the time series belong to turning process and their frequency domain feature matrices computed in Matlab as input. It perfoms the 2 class classifications with four different classifiers which user should specify in parameters section.

WPT_Feature_Extraction.**WPT_Feature_Extraction**(*stickout_length*, *WPT_Level*, *Classifier*, *plotting*)

Parameters

- **stickout_length** – The distance between heel of the boring bar and the back surface of the cutting tool
  - if stickout length is 2 inch, '2'
  - if stickout length is 2.5 inch, '2p5'
  - if stickout length is 3.5 inch, '3p5'
  - if stickout length is 4.5 inch, '4p5'
- **WPT_Level** – Level of Wavelet Packet Decomposition
- **Classifier** – Classifier defined by user
  - Support Vector Machine: 'SVC'
  - Logistic Regression: 'LR'
  - Random Forest Classification: 'RF'
  - Gradient Boosting: 'GB'
- **Plotting** – Function will return the plot of the results depending on the number of features used in the classification, if the Plotting is 'True'.

Returns

results Classification results for training and test set for all combination of ranked features

time Elapsed time during feature matrix generation and classification

Example

```
>>> from WPT_Feature_Extraction import WPT_Feature_Extraction
>>> import matplotlib.pyplot as plt
>>> from matplotlib import rc
>>> import matplotlib
```

(continues on next page)

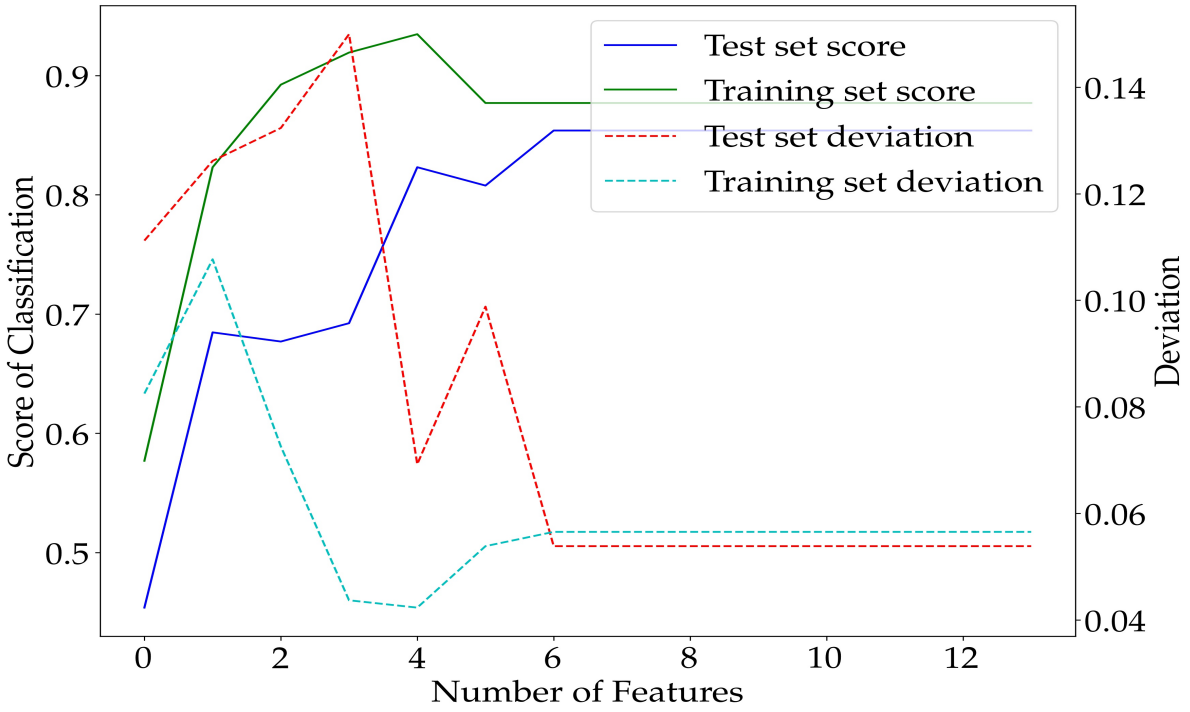
(continued from previous page)

```
>>> matplotlib.rcParams.update({'font.size': 14})
>>> rc('font',**{'family':'serif','serif':['Palatino']})
>>> rc('text', usetex=True)

#parameters

>>> stickout_length='2'
>>> WPT_Level = 4
>>> Classifier = 'SVC'
>>> plotting = True

>>> results = WPT_Feature_Extraction(stickout_length, WPT_Level,
>>>                                  Classifier, plotting)
Enter the path of the data files:
>>> D\...\cutting_tests_processed\data_2inch_stickout
```



1.2 Transfer Learning Application Using WPT

This fuction takes the reconstructed time series after WPT and their freuqency domain features as input. The time domain features are computed inside of the function. Since this algorithm uses transfer learning principle, user needs to specify the stickout length of the training set and test set data. The function returns classification results in array for both test set and training set.

`WPT_Transfer_Learning.`**`WPT_Transfer_Learning`**(*stickout_length_training*, *stickout_length_test*, *WPT_Level*, *Classifier*)

Parameters

- **`stickout_length_training`** – Stickout length for the training data set
  - if stickout length is 2 inch, '2'

  - if stickout length is 2.5 inch, '2p5'
  - if stickout length is 3.5 inch, '3p5'
  - if stickout length is 4.5 inch, '4p5'
- **stickout_length_test** – Stickout length for the test data set
  - if stickout length is 2 inch, '2'
  - if stickout length is 2.5 inch, '2p5'
  - if stickout length is 3.5 inch, '3p5'
  - if stickout length is 4.5 inch, '4p5'
- **WPT_Level** – Level of Wavelet Packet Decomposition
- **Classifier** –

  Classifier defined by user

  - Support Vector Machine: 'SVC'
  - Logistic Regression: 'LR'
  - Random Forest Classification: 'RF'
  - Gradient Boosting: 'GB'

Returns

results Classification results for training and test set for all combination of ranked features

time Elapsed time during feature matrix generation and classification

Example

```
>>> from WPT_Transfer_Learning import WPT_Transfer_Learning

#parameters

>>> stickout_length_training = '2'
>>> stickout_length_test = '4p5'
>>> WPT_Level=4
>>> Classifier='SVC'

>>> results = WPT_Transfer_Learning(stickout_length_training,
>>>                                  stickout_length_test,
>>>                                 WPT_Level, Classifier)
Enter the path of training set data files:
>>> D\...\cutting_tests_processed\data_2inch_stickout
Enter the path of test set data files:
>>> D\...\cutting_tests_processed\data_4p5inch_stickout
```

1.3 Transfer Learning On Two Cases Using WPT

This fuction implement transfer learning by training a classifier on two different data sets and testing it on remaining two different data sets. Stickout length of each data set should be determined by user.

WPT_Transfer_Learning_2case.**WPT_Transfer_Learning_2case**(*stickout_lengths*, *WPT_Level*, *Classifier*)

Parameters

- **stickout_lengths** – Stickout length for the training and test set in a np.array([]) format.First two stickout length are considered as training set data and the remaining ones are test set data.
  - if stickout length is 2 inch, '2'
  - if stickout length is 2.5 inch, '2p5'
  - if stickout length is 3.5 inch, '3p5'
  - if stickout length is 4.5 inch, '4p5'
- **WPT_Level** – Level of Wavelet Packet Decomposition
- **Classifier** –

  Classifier defined by user

  - Support Vector Machine: 'SVC'
  - Logistic Regression: 'LR'
  - Random Forest Classification: 'RF'
  - Gradient Boosting: 'GB'

Returns

results Classification results for training and test set for all combination of ranked features

time Elapsed time during feature matrix generation and classification

Example

```
>>> from WPT_Transfer_Learning_2case import WPT_Transfer_Learning_
↪2case

#parameters

>>> stickout_lengths = ['2','2p5','3p5','4p5']
>>> WPT_Level=4
>>> Classifier='SVC'

>>> results = WPT_Transfer_Learning_2case(stickout_lengths,
>>>                                       WPT_Level, Classifier)
Enter the path of first training set data files:
>>> D\...\cutting_tests_processed\data_2inch_stickout
Enter the path of second training set data files:
>>> D\...\cutting_tests_processed\data_2p5inch_stickout
Enter the path of first test set data files:
>>> D\...\cutting_tests_processed\data_3p5inch_stickout
Enter the path of second test set data files:
>>> D\...\cutting_tests_processed\data_4p5inch_stickout
```

CHAPTER

TWO

ENSEMBLE EMPIRICAL MODE DECOMPOSITION (EEMD)

2.1 Feature extraction and supervised classification using EEMD

This function takes time series and decompose them using Ensemble Empirical Mode Decomposition (EEMD). If user already computed the decompositions, algorithm will directly compute the features and generate feature matrices. Algorithm require user give classifier name, informative intrinsic mode function (IMF) number and the stickout length of the data user working on.

EEMD_Feature_Extraction.**EEMD_Feature_Extraction**(*stickout_length*, *EEMDecs*, *p*, *Classifier*)

Parameters

- **stickout_length** – The distance between heel of the boring bar and the back surface of the cutting tool
  - if stickout length is 2 inch, '2'
  - if stickout length is 2.5 inch, '2p5'
  - if stickout length is 3.5 inch, '3p5'
  - if stickout length is 4.5 inch, '4p5'
- **EEMDecs** –
  - if decompositions have already been computed, 'A'
  - if decompositions have not been computed, 'NA'
- **p** – Informative intrinsic mode function (IMF) number
- **Classifier** – Classifier defined by user
  - Support Vector Machine: 'SVC'
  - Logistic Regression: 'LR'
  - Random Forest Classification: 'RF'
  - Gradient Boosting: 'GB'

Returns

results Classification results for training and test set for all combination of ranked features

time Elapsed time during feature matrix generation and classification

Example

```
>>> from EEMD_Feature_Extraction import EEMD_Feature_Extraction

#parameters
>>> stickout_length='2'
>>> EEMDecs = 'A'
>>> p=2
>>> Classifier = 'SVC'

>>> results = WPT_Feature_Extraction(stickout_length, WPT_Level,
>>>                                  Classifier, plotting)
Enter the path of the data files:
>>> D\...\cutting_tests_processed\data_2inch_stickout
Enter Enter the path of EEMD files:
>>> D\...\cutting_tests_processed\data_2inch_stickout
```

2.2 Transfer Learning Application Using EEMD

This fuction uses tranfer learning principles to diagnose chatter in time series obtained from turning cutting experiment. Intrinsic mode functions (IMFs) or decomposition for each time series should have been computed to be able to use this code.

EEMD_Transfer_Learning.**EEMD_Transfer_Learning**(*stickout_length_training*, *stickout_length_test*, *p_train*, *p_test*, *Classifier*)

Parameters

- **stickout_length_training** – Stickout length for the training data set
  - if stickout length is 2 inch, '2'
  - if stickout length is 2.5 inch, '2p5'
  - if stickout length is 3.5 inch, '3p5'
  - if stickout length is 4.5 inch, '4p5'
- **stickout_length_test** – Stickout length for the test data set
  - if stickout length is 2 inch, '2'
  - if stickout length is 2.5 inch, '2p5'
  - if stickout length is 3.5 inch, '3p5'
  - if stickout length is 4.5 inch, '4p5'
- **p_train** – Informative intrinsic mode function (IMF) number for training set
- **p_test** – Informative intrinsic mode function (IMF) number for test set
- **Classifier** – Classifier defined by user
  - Support Vector Machine: 'SVC'
  - Logistic Regression: 'LR'
  - Random Forest Classification: 'RF'
  - Gradient Boosting: 'GB'

Returns

results Classification results for training and test set for all combination of ranked features

time Elapsed time during feature matrix generation and classification

Example

```
>>> from EEMD_Feature_Extraction import EEMD_Feature_Extraction

#parameters
>>> stickout_length_training='2'
>>> stickout_length_test='4p5'
>>> p_train = 2
>>> p_test = 1
>>> Classifier = 'GB'

>>> results = EEMD_Transfer_Learning(stickout_length_training,
                                     stickout_length_test,
>>>                                  p_train, p_test,
                                     Classifier)
Enter the path of training data files:
>>> D\...\cutting_tests_processed\data_2inch_stickout
Enter the path of test data files:
>>> D\...\cutting_tests_processed\data_4p5inch_stickout
Enter the path to decompositions for training set:
>>> D\...\eIMFs\data_2inch_stickout
Enter the path to decompositions for test set:
>>> D\...\eIMFs\data_4p5inch_stickout
```

CHAPTER

THREE

REFERENCES

- genindex
- modindex
- search

BIBLIOGRAPHY

[1] Melih C. Yesilli, Firas A. Khasawneh, and Andreas Otto. On transfer learning for chatter detection in turning using wavelet packet transform and ensemble empirical mode decomposition. *CIRP Journal of Manufacturing Science and Technology*, dec 2019. doi:10.1016/j.cirpj.2019.11.003.

PYTHON MODULE INDEX

e

w

INDEX

E

W