MALT

Release 1.0

Oct 15, 2019

Contents:

1 How to install MALT **3**

2 Example Notebooks **5**
- 2.1 Lightcurve Example . . . 5
- 2.2 Dataset Example . . . 5

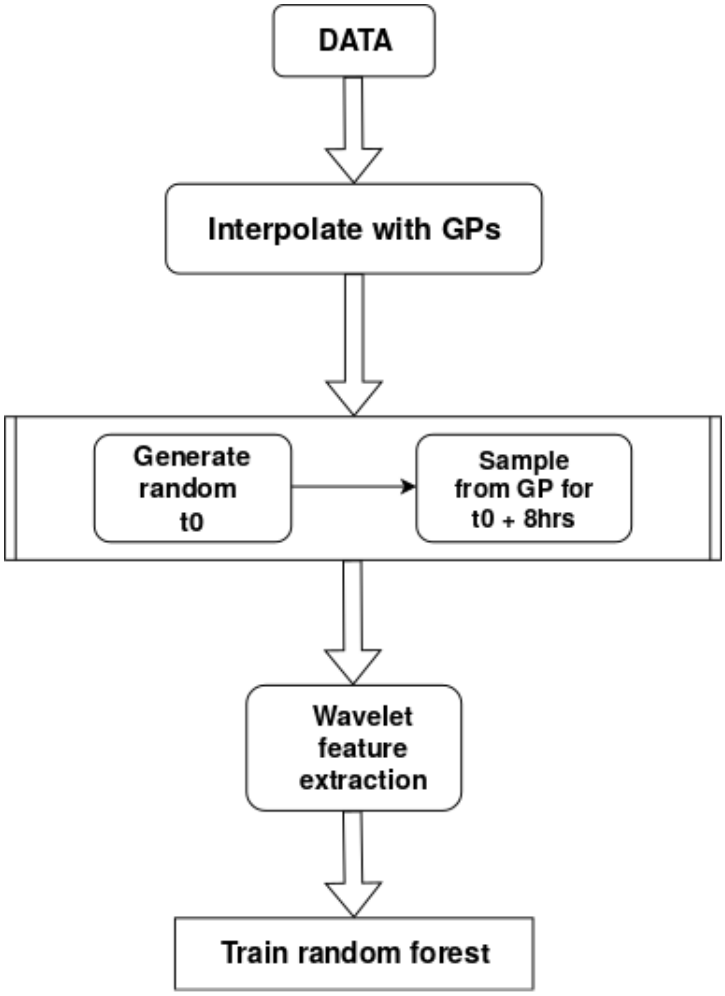
3 The MALT API reference **7**
- 3.1 The Lightcurve class . . . 7
- 3.2 The Dataset class . . . 8
- 3.3 MALT interpolator . . . 9
- 3.4 MALT feature extraction . . . 9
- 3.5 MALT machine learning . . . 9

Python Module Index **11**

Index **13**

MALT is a classification pipeline based on the paper "Classification of Multiwavelength Transients with Machine Learning" by Sooknunan et al. (2018). It is a framework which allows the user to classify time series data. The user is free to choose the interpolation technique, feature extraction method, and the machine learning classifier to use.

The default pipeline is shown below. It uses Gaussian processes to interpolate the data, a wavelet feature extraction method and a random forest classifier.



CHAPTER 1

How to install MALT

First clone the git repo and install virtualenv if not already installed:

```
git clone https://github.com/kimeels/MALT.git

python3 -m pip install --user virtualenv
```

Change directories into MALT and create a virtual environment:

```
cd MALT

python3 -m venv malt_env
```

Start the virtual env and install the necessary packages using the requirements file:

```
source malt_env/bin/activate

pip3 install -r requirements.txt
```

CHAPTER 2

Example Notebooks

Install Jupyter Notebooks/Labs if not already installed.

2.1 Lightcurve Example

Change directories into Examples and start the MALT_lightcurve_class_example notebook:

```
cd MALT/Examples
jupyter notebook MALT_lightcurve_class_example notebook.ipynb
```

2.2 Dataset Example

Change directories into Examples and start the MALT_dataset_class_example notebook:

```
cd MALT/Examples
jupyter notebook MALT_dataset_class_example notebook.ipynb
```

CHAPTER 3

The MALT API reference

3.1 The Lightcurve class

class malt.**Lightcurve**(*filepath*, *interpolate=False*, *interp_func=<function get_gp>*, *ini_t='rand'*, *obs_time=0.3333333333333333*, *sample_size=100*, *obj_type=None*)

extract_features(*feat_ex_method=<function get_wavelet_feature>*)

Extracts features from the given lightcurve with assigned feature extraction method.

self: Lightcurve object An instance of the Lightcurve class.

feat_ex_method: python function Function to use for the feature extraction.

interpolate(*interp_func=<function get_gp>*, *ini_t='rand'*, *obs_time=0.3333333333333333*, *sample_size=100*, *aug_num=1*)

Interpolates the given lightcurve with assigned interpolation function

self: Lightcurve object An instance of the Lightcurve class.

interp_func: python function A python function that takes in a lightcurve and interpolates it.

ini_t: str or float Initial time to start sampling.

obs_time: float The total length of the interpolated lightcurve.

sample_size: int Number of data points in interpolated lightcurve.

aug_num: int Number of lightcurves to augment to.

loadfile(*filename*)

Loads file to extract time, flux, flux_err ra_dec and class

filename: path to dataset

3.2 The Dataset class

class malt.**Dataset**(*configFile=”, feat_ex_method=<function get_wavelet_feature>, interpolate=True, interp_func=<function get_gp>, ini_t=’rand’, obs_time=0.3333333333333333, sample_size=100, aug_num=1, ml_method=<class ’malt.machine_learning.RFclassifier’>, hyperparams={’criterion’: [’gini’, ’entropy’], ’n_estimators’: array([70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89])}, n_jobs=-1, pca=True, n_components=20*)

add(*new_lightcurve*)

Adds new lightcurve to the Dataset then retrains Dataset.

self: Dataset object An instance of the Dataset class containing instances of the Lightcurve class.

lightcurve: Lightcurve object Lightcurve object to add to dataset.

extract_features()

Extracts features from all the lightcurves in the given dataset with assigned feature extraction method.

self: Dataset object An instance of the Dataset class containing instances of the Lightcurve class.

get_pca()

Performs PCA decomposition of a feature array X.

self: Dataset object An instance of the Dataset class containing instances of the Lightcurve class.

interpolate()

Interpolates all the lightcurves in the given dataset with assigned interpolation function.

self: Dataset object An instance of the Dataset class containing instances of the Lightcurve class.

classmethod load_from_save(*filename*)

Returns a saved Dataset instance using pickle

self: Dataset object An instance of the Dataset class containing instances of the Lightcurve class.

filename: str filename under which the Dataset instance was saved.

populate(*filepaths*)

Initialises an instance of the Dataset class.

self: Database object An instance of the Database class.

filepaths: list List containing the paths to the data files.

predict(*lightcurve, show_prob=False*)

Predicts the type of given lightcurve object using classifier trained on Dataset.

self: Dataset object An instance of the Dataset class containing instances of the Lightcurve class.

lightcurve: Lightcurve object Lightcurve object for which to predict

show_prob: boolean. If True will print full output from predict_proba()

project_pca(*lightcurve=None*)

Projects self.features onto calculated PCA axis from self.pca

self: Dataset object An instance of the Dataset class containing instances of the Lightcurve class.

run_diagnostic()

Runs the Diagnostic test which trains n classifiers on different subsets of the Dataset to test how well it can classify objects.

self: Dataset object An instance of the Dataset class containing instances of the Lightcurve class.

save (*filename='saved_dataset'*)

Saves a Dataset instance using a pickle dump

self: Dataset object An instance of the Dataset class containing instances of the Lightcurve class.

filename: str filename under which to store the Dataset instance

train (*verbose=1*)

Trains a ML algorithm on the Dataset with the parameters specified on initialisation.

self: Dataset object An instance of the Dataset class containing instances of the Lightcurve class.

verbose: How much information to print out.

types (*show_aug_num=False*)

Prints out the counts of each object type stored in the dataset.

self: Dataset object An instance of the Dataset class containing instances of the Lightcurve class.

show_aug_num: boolean Use augmented lightcurve when counting type numbers.

3.3 MALT interpolator

malt.interpolator.**get_gp** (*lightcurve, t0, obs_time, sample_size, aug_num*)

Returns a Gaussian Process (george) object marginalised on the data in file.

lightcurve: Lightcurve object An instance of the Lightcurve class.

t0: float Initial time to start sampling.

obs_time: float The total length of the interpolated lightcurve.

sample_size: int Number of data points in interpolated lightcurve.

3.4 MALT feature extraction

malt.feature_extraction.**get_wavelet_feature** (*lightcurve*)

Returns wavelet coefficients for a given lightcurve object.

lightcurve: Lightcurve object An instance of the Lightcurve class

3.5 MALT machine learning

class malt.machine_learning.**RFclassifier** (*n_estimators='warn', criterion='gini'*)

Python Module Index

m

malt.feature_extraction, 9
malt.interpolator, 9
malt.machine_learning, 9

Index

A

`add()` (*malt.Dataset method*), 8

D

`Dataset` (*class in malt*), 8

E

`extract_features()` (*malt.Dataset method*), 8
`extract_features()` (*malt.Lightcurve method*), 7

G

`get_gp()` (*in module malt.interpolator*), 9
`get_pca()` (*malt.Dataset method*), 8
`get_wavelet_feature()` (*in module malt.feature_extraction*), 9

I

`interpolate()` (*malt.Dataset method*), 8
`interpolate()` (*malt.Lightcurve method*), 7

L

`Lightcurve` (*class in malt*), 7
`load_from_save()` (*malt.Dataset class method*), 8
`loadfile()` (*malt.Lightcurve method*), 7

M

`malt.feature_extraction` (*module*), 9
`malt.interpolator` (*module*), 9
`malt.machine_learning` (*module*), 9

P

`populate()` (*malt.Dataset method*), 8
`predict()` (*malt.Dataset method*), 8
`project_pca()` (*malt.Dataset method*), 8

R

`RFclassifier` (*class in malt.machine_learning*), 9
`run_diagnostic()` (*malt.Dataset method*), 8

S

`save()` (*malt.Dataset method*), 9

T

`train()` (*malt.Dataset method*), 9
`types()` (*malt.Dataset method*), 9