GENDIS Documentation

Gilles Vandewiele

Sep 03, 2018

Contents

1 Installation **3**

2 Getting Started **5**

2.1 1. Loading & preprocessing the datasets 5

2.2 2. Creating a `GeneticExtractor` object 5

2.3 3. Fit the `GeneticExtractor` and construct distance matrix 6

2.4 4. Fit ML classifier on constructed distance matrix 6

3 GENDIS **7**

4 Contributing, Citing and Contact **9**

In the time series classification domain, shapelets are small subseries that are discriminative for a certain class. It has been shown that by projecting the original dataset to a distance space, where each axis corresponds to the distance to a certain shapelet, classifiers are able to achieve state-of-the-art results on a plethora of datasets.

This repository contains an implementation of `GENDIS`, an algorithm that searches for a set of shapelets in a genetic fashion. The algorithm is insensitive to its parameters (such as population size, crossover and mutation probability, . . .) and can quickly extract a small set of shapelets that is able to achieve predictive performances similar (or better) to that of other shapelet techniques.

CHAPTER 1

Installation

We currently support Python 3.5 & Python 3.6. For installation, there are two alternatives:

1. Clone the repository `https://github.com/IBCNServices/GENDIS.git` and run `(python3 -m) pip -r install requirements.txt`
2. GENDIS is hosted on PyPi. You can just run `(python3 -m) pip install gendis` to add gendis to your dist-packages (you can use it from everywhere).

CHAPTER 2

Getting Started

2.1 1. Loading & preprocessing the datasets

In a first step, we need to construct at least a matrix with timeseries (`X_train`) and a vector with labels (`y_train`). Additionally, test data can be loaded as well in order to evaluate the pipeline in the end.

```
import pandas as pd
# Read in the datafiles
train_df = pd.read_csv(<DATA_FILE>)
test_df = pd.read_csv(<DATA_FILE>)
# Split into feature matrices and label vectors
X_train = train_df.drop('target', axis=1)
y_train = train_df['target']
X_test = test_df.drop('target', axis=1)
y_test = test_df['target']
```

2.2 2. Creating a `GeneticExtractor` object

Construct the object. For a list of all possible parameters, and a description, please refer to the documentation in the code

```
from gendis.genetic import GeneticExtractor
genetic_extractor = GeneticExtractor(population_size=50, iterations=25, verbose=False,
                                     normed=False, add_noise_prob=0.3, add_shapelet_
↪prob=0.3,
                                     wait=10, plot='notebook', remove_shapelet_prob=0.
↪3,
                                     crossover_prob=0.66, n_jobs=4)
```

2.3 3. Fit the `GeneticExtractor` and construct distance matrix

```
shapelets = genetic_extractor.fit(X_train, y_train)
distances_train = genetic_extractor.transform(X_train)
distances_test = genetic_extractor.transform(X_test)
```

2.4 4. Fit ML classifier on constructed distance matrix

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
lr = LogisticRegression()
lr.fit(distances_train, y_train)

print('Accuracy = {}'.format(accuracy_score(y_test, lr.predict(distances_test))))
```

CHAPTER 3

GENDIS

class gendis.genetic.**GeneticExtractor**(*population_size=50*, *iterations=25*, *verbose=False*, *normed=False*, *add_noise_prob=0.4*, *add_shapelet_prob=0.4*, *wait=10*, *plot=None*, *remove_shapelet_prob=0.4*, *crossover_prob=0.66*, *n_jobs=4*)

Feature selection with genetic algorithm.

population_size [int] The number of individuals in our population. Increasing this parameter increases both the runtime per generation, as the probability of finding a good solution.

iterations [int] The maximum number of generations the algorithm may run.

wait [int] If no improvement has been found for *wait* iterations, then stop

add_noise_prob [float] The chance that gaussian noise is added to a random shapelet from a random individual every generation

add_shapelet_prob [float] The chance that a shapelet is added to a random shapelet set every gen

remove_shapelet_prob [float] The chance that a shapelet is deleted to a random shapelet set every gen

crossover_prob [float] The chance that of crossing over two shapelet sets every generation

normed [boolean] Whether we first have to normalize before calculating distances

n_jobs [int] The number of threads to use

verbose [boolean] Whether to print some statistics in every generation

plot [object] Whether to plot the individuals every generation (if the population size is smaller than or equal to 20), or to plot the fittest individual

shapelets [array-like] The fittest shapelet set after evolution

label_mapping: dict A dictionary that maps the labels to the range [0, . . . , C-1]

An example showing genetic shapelet extraction on a simple dataset:

```
>>> from tslearn.generators import random_walk_blobs
>>> from genetic import GeneticExtractor
>>> from sklearn.linear_model import LogisticRegression
>>> import numpy as np
>>> np.random.seed(1337)
>>> X, y = random_walk_blobs(n_ts_per_blob=20, sz=64, noise_level=0.1)
>>> X = np.reshape(X, (X.shape[0], X.shape[1]))
>>> extractor = GeneticExtractor(iterations=5, n_jobs=1, population_size=10)
>>> distances = extractor.fit_transform(X, y)
>>> lr = LogisticRegression()
>>> _ = lr.fit(distances, y)
>>> lr.score(distances, y)
1.0
```

Methods

`__init__`([population_size, iterations, …])	Initialize self.
`fit`(X, y)	Extract shapelets from the provided timeseries and labels.
`transform`(X)	After fitting the Extractor, we can transform collections of timeseries in matrices with distances to each of the shapelets in the evolved shapelet set.
`fit_transform`(X, y)	Combine both the fit and transform method in one.
`save`(path)	Write away all hyper-parameters and discovered shapelets to disk
`load`(path)	Instantiate a saved GeneticExtractor

CHAPTER 4

Contributing, Citing and Contact

For now, please refer to this repository. A paper, to which you can then refer, will be published in the nearby future. If you have any questions, are experiencing bugs in the GENDIS implementation, or would like to contribute, please feel free to create an issue/pull request in this repository or take contact with me at gilles(dot)vandewiele(at)ugent(dot)be

Index

G

GeneticExtractor (class in gendis.genetic), 7