sktime.transformers.shapelets module

class
sktime.transformers.shapelets.ContractedRandomShapeletTransform(*min_shapelet_length=3, max_shapelet_length=inf, initial_num_shapelets_per_case=15, time_limit_in_mins=60.0, seed=0, dim_to_use=0, remove_self_similar=True, verbose=False, use_binary_info_gain=True, trim_shapelets=True, num_shapelets_to_trim_to=200*) [source]

Bases: `sktime.transformers.shapelets.RandomShapeletTransform`

Contracted Random Shapelet Transform (extends Random Shapelet Transform).

A transformer to extract shapelets from a training dataset, then transform any passed dataset into primitives using each extracted shapelet. For k shapelets, an input case q is transformed by calculating the distance from q to each shapelet and the transformed output case is composed of k features, where the kth attribute is the distance from q to the kth shapelet.

This implementation of the transform is based on the findings in [2]; it uses random sampling of candidate shapelets, rather than a full enumeration of candidates as originally proposed in [1], as it has been shown that there is no significant decrease in accuracy but a significant reduction in runtime through this approach.

There are two versions of the transform: RandomShapeletTransform and ContractedRandomShapeletTransform. For each training series visited, both implementations assess a specified number of candidate shapelets per series. However, this version uses a contracted time limit; it visits a series, extracts an explict number of shapelets, and then moves onto the next series. This continues until the time limit is breached, and if the end of the data is reached first, the algorithm will loop back round to the first series and extract the specified number of shapelets again. The time limit is specified in minutes and is a lower-bound; if a shapelet is being evaluated when the time limit is reached, it will complete before the extraction process exits.

Parameters

- **min_shapelet_length** (*int (default = 3)*) – The minimum candidate shapelet length
- **max_shapelet_length** (*int (default = np.inf)*) – The maximum candidate shapelet length (default = np.inf). This value is an upper-bound and is automatically capped to the series length if the specified value exceeds the length of the shapelet.
- **time_limit_in_mins** (*float (default = 60.0)*) – The contract time limit to continue shapelet extraction. This is a lower-bound as a shapelet candidate will not be abandoned if the time limit runs out.

- **initial_num_shapelets_per_case** (*int (default = 15)*) – The number of shapelet candidates to evaluate per series in the fit method initially. If there is sufficient time remaining on the contract after this number of shapelets have been evaluated from all training series, the search will continue at the start of the data and extract this many shapelets from each series again.
- **seed** (*int (default = 0)*) – To seed the shapelet discovery to ensure deterministic results across multiple runs
- **dim_to_use** (*int (default = 0)*) – Which dimension of the data to use
- **remove_self_similar** (*bool (default = True)*) – Whether to remove self-similar shapelets before transforming. A candidate shapelet is considered to be self-similar to another candidate if they are taken from the same dimension of the same training series. With this set to True, any overlapping candidates will be removed to preserve the best candidate (i.e. highest info gain)
- **verbose** (*bool (default = False)*) – Whether to print information during shapelet extraction
- **use_binary_info_gain** (*bool (default = True)*) – Whether to use one-vs-all information gain (as in [2]) or consider each class independently when calculating the information gain of a candidate (as in [1])
- **trim_shapelets** (*bool (default = True)*) – Whether to crop the final list of extracted shapelets to the top k. If specified, this is an upper-bound; if less than k shapelets are found then this parameter will have no effect
- **num_shapelets_to_trim_to** (*int (default = 200)*) – If trim_shapelets = True, this is the number of shapelets to trim to.

shapelets

The list of extracted shapelets after fit is called (initially shapelets = None)

Type

list of Shapelet objects

References

[1] Hills, Jon, Jason Lines, Edgaras Baranauskas, James Mapp, and Anthony Bagnall. “Classification of time series by shapelet transformation.” Data Mining and Knowledge Discovery 28, no. 4 (2014): 851-881.

[2] Bostrom, Aaron, and Anthony Bagnall. “Binary shapelet transform for multiclass time series classification.” In Transactions on Large-Scale Data-and Knowledge-Centered Systems XXXII, pp. 24-46. Springer, Berlin, Heidelberg, 2017.

class **sktime.transformers.shapelets.RandomShapeletTransform**(*min_shapelet_length=3, max_shapelet_length=inf, num_cases_to_sample=5, num_shapelets_to_sample_per_case=15, seed=0,*

dim_to_use=0, remove_self_similar=True, verbose=False, use_binary_info_gain=True, trim_shapelets=True, num_shapelets_to_trim_to=200) [source]

Bases: `sktime.transformers.base.BaseTransformer`

Random Shapelet Transform.

A transformer to extract shapelets from a training dataset, then transform any passed dataset into primitives using each extracted shapelet. For k shapelets, an input case q is transformed by calculating the distance from q to each shapelet and the transformed output case is composed of k features, where the kth attribute is the distance from q to the kth shapelet.

This implementation of the transform is based on the findings in [2]; it uses random sampling of candidate shapelets, rather than a full enumeration of candidates as originally proposed in [1], as it has been shown that there is no significant decrease in accuracy but a significant reduction in runtime through this approach.

There are two versions of the transform: RandomShapeletTransform and ContractedRandomShapeletTransform. For each training series visited, both implementations assess a specified number of candidate shapelets per series. However, this version visits an explict number of cases to extract these shapelets from (from 1 to len(X), whereas ContractedRandomShapeletTransform visits training cases while time remains on a contact (specified in minutes).

Parameters

- **min_shapelet_length** (*int (default = 3)*) – The minimum candidate shapelet length
- **max_shapelet_length** (*int (default = np.inf)*) – The maximum candidate shapelet length (default = np.inf). This value is an upper-bound and is automatically capped to the series length if the specified value exceeds the length of the shapelet.
- **num_cases_to_sample** (*int (default = 5)*) – The number of training cases to extract candidate shapelets from.
- **num_shapelets_to_sample_per_case** (*int (default = 15)*) – The number of shapelet candidates to evaluate per series in the fit method
- **seed** (*int (default = 0)*) – To seed the shapelet discovery to ensure deterministic results across multiple runs
- **dim_to_use** (*int (default = 0)*) – Which dimension of the data to use
- **remove_self_similar** (*bool (default = True)*) – Whether to remove self-similar shapelets before transforming. A candidate shapelet is considered to be self-similar to another candidate if they are taken from the same dimension of the same training series. With this set to True, any overlapping candidates will be removed to preserve the best candidate (i.e. highest info gain)
- **verbose** (*bool (default = False)*) – Whether to print information during shapelet extraction

- **use_binary_info_gain** (*bool (default = True)*) – Whether to use one-vs-all information gain (as in [2]) or consider each class independently when calculating the information gain of a candidate (as in [1])
- **trim_shapelets** (*bool (default = True)*) – Whether to crop the final list of extracted shapelets to the top k. If specified, this is an upper-bound; if less than k shapelets are found then this parameter will have no effect
- **num_shapelets_to_trim_to** (*int (default = 200)*) – If trim_shapelets = True, this is the number of shapelets to trim to.

shapelets

The list of extracted shapelets after fit is called (initially shapelets = None)

Type

list of Shapelet objects

[1] Hills, Jon, Jason Lines, Edgaras Baranauskas, James Mapp, and Anthony Bagnall. "Classification of time series by shapelet transformation." Data Mining and Knowledge Discovery 28, no. 4 (2014): 851-881.

[2] Bostrom, Aaron, and Anthony Bagnall. "Binary shapelet transform for multiclass time series classification." In Transactions on Large-Scale Data-and Knowledge-Centered Systems XXXII, pp. 24-46. Springer, Berlin, Heidelberg, 2017.

static **calc_info_gain**(*orderline*, *class_counts*, *total*) [source]

A static method to calculate the information gain of a candidate shapelet when given an orderline and class distribution of distances to a dataset.

Parameters

orderline (*pandas DataFrame*) – The input dataframe to transform

Returns

output – The transformed dataframe in tabular format.

Return type

pandas DataFrame

fit(*X*, *y*, ***fit_params*) [source]

A method to fit the shapelet transform to a specified X and y

Parameters

- **X** (*pandas DataFrame*) – The training input samples.
- **y** (*array-like or list*) – The class values for X

Returns

self – This estimator

Return type

RandomShapeletTransform

fit_transform(*X*, *y=None*, ***fit_params*) [source]

Fits and transforms a given input X and y

Parameters

- **X** (*pandas.DataFrame the input data to transform*) –
- **y** (*list or array like of class values corresponding to the indices in X*) –

Returns

Xt – The transformed pandas DataFrame.

Return type

pandas DataFrame

get_shapelets() [source]

An accessor method to return the extracted shapelets

Returns

shapelets

Return type

a list of Shapelet objects

static **remove_self_similar**(*shapelet_list*) [source]

Remove self-similar shapelets from an input list. Note: this method assumes that shapelets are pre-sorted in descending order of quality (i.e. if two candidates are self-similar, the one with the later index will be removed)

Parameters

shapelet_list (*list of Shapelet objects*) –

Returns

shapelet_list

Return type

list of Shapelet objects

transform(*X*, *y=None*, ***transform_params*) [source]

Transforms X according to the extracted shapelets (self.shapelets)

Parameters

X (*pandas DataFrame*) – The input dataframe to transform

Returns

output – The transformed dataframe in tabular format.

Return type

pandas DataFrame

static **zscore**(*a*, *axis=0*, *ddof=0*) [source]

A static method to return the normalised version of series. This mirrors the scipy implementation with a small difference - rather than allowing /0, the function returns output = np.zeroes(len(input)). This is to allow for sensible processing of candidate shapelets/comparison subseries that are a straight line. Original version: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.zscore.html

Parameters

- **a** (*array_like*) – An array like object containing the sample data.
- **axis** (*int or None*, *optional*) – Axis along which to operate. Default is 0. If None, compute over the whole array a.
- **ddof** (*int*, *optional*) – Degrees of freedom correction in the calculation of the standard deviation. Default is 0.

Returns

zscore – The z-scores, standardized by mean and standard deviation of input array a.

Return type

array_like

---

class **sktime.transformers.shapelets.Shapelet**(*series_id*, *start_pos*, *length*, *info_gain*, *data*) [source]

Bases: `object`

A simple class to model a Shapelet with associated information

Parameters

- **series_id** (*int*) – The index of the series within the data (X) that was passed to fit.
- **start_pos** (*int*) – The starting position of the shapelet within the original series
- **length** (*int*) – The length of the shapelet

- **info_gain** (*flaot*) – The calculated information gain of this shapelet
- **data** (*array-like*) – The (z-normalised) data of this shapelet