
Sliceline

Antoine de Daran

Apr 11, 2023

CONTENTS

1	Getting started	3
2	Installation	5
3	Useful links	7
4	Contributing	9
5	License	11
5.1	Slicefinder	11
	Index	13

Sliceline is a Python library for fast slice finding for Machine Learning model debugging.

It is an implementation of [SliceLine: Fast, Linear-Algebra-based Slice Finding for ML Model Debugging](#), from Svetlana Sagadeeva and Matthias Boehm of Graz University of Technology.

GETTING STARTED

Given an input dataset **X** and a model error vector **errors**, SliceLine finds the top slices in **X** that identify where a ML model performs significantly worse.

You can use sliceline as follows:

```
from sliceline.slicefinder import Slicefinder
slice_finder = Slicefinder()
slice_finder.fit(X, errors)
print(slice_finder.top_slices_)
X_trans = slice_finder.transform(X)
```

We invite you to check the [demo notebooks](#) for a more thorough tutorial:

1. Implementing Sliceline on Titanic dataset
2. Implementing Sliceline on California housing dataset

INSTALLATION

Sliceline is intended to work with **Python 3.7 or above**. Installation can be done with `pip`:

```
pip install sliceline
```

There are [wheels available](#) for Linux, MacOS, and Windows, which means that you most probably won't have to build Sliceline from source.

You can install the latest development version from GitHub as so:

```
pip install git+https://github.com/DataDome/sliceline --upgrade
```

Or, through SSH:

```
pip install git+ssh://git@github.com/datadome/sliceline.git --upgrade
```


USEFUL LINKS

- [Documentation](#)
- [Package releases](#)
- [SliceLine paper](#)

CONTRIBUTING

Feel free to contribute in any way you like, we're always open to new ideas and approaches.

- [Open a discussion](#) if you have any question or enquiry whatsoever. It's more useful to ask your question in public rather than sending us a private email. It's also encouraged to open a discussion before contributing, so that everyone is aligned and unnecessary work is avoided.
- Feel welcome to [open an issue](#) if you think you've spotted a bug or a performance issue.

Please check out the [contribution guidelines](#) if you want to bring modifications to the code base.

LICENSE

Sliceline is free and open-source software licensed under the [3-clause BSD license](#).

5.1 Slicefinder

```
class sliceline.Slicefinder(alpha: float = 0.6, k: int = 1, max_l: int = 4, min_sup: int | float = 10, verbose: bool = True)
```

Slicefinder class.

SliceLine is a fast, linear-algebra-based slice finding for ML Model Debugging.

Given an input dataset (X) and a model error vector ($errors$), SliceLine finds the k slices in X that identify where the model performs significantly worse. A slice is a subspace of X defined by one or more predicates. The maximal dimension of this subspace is controlled by max_l .

The slice scoring function is the linear combination of two objectives:

- Find sufficiently large slices, with more than min_sup elements (high impact on the overall model)
- With substantial errors (high negative impact on sub-group/model)

The importance of each objective is controlled through a single parameter $alpha$.

Slice enumeration and pruning techniques are done via sparse linear algebra.

5.1.1 Parameters

alpha: float, default=0.6

Weight parameter for the importance of the average slice error. $0 < alpha \leq 1$.

k: int, default=1

Maximum number of slices to return. Note: in case of equality between k -th slice score and the following ones, all those slices are returned, leading to $_n_features_out$ slices returned. ($_n_features_out \geq k$)

max_l: int, default=4

Maximum lattice level. In other words: the maximum number of predicate to define a slice.

min_sup: int or float, default=10

Minimum support threshold. Inspired by frequent itemset mining, it ensures statistical significance. If min_sup is a float ($0 < min_sup < 1$),

it represents the fraction of the input dataset (X).

verbose: bool, default=True

Controls the verbosity.

5.1.2 Attributes

top_slices_: np.ndarray of shape (`_n_features_out`, number of columns of the input dataset)

The `_n_features_out` slices with the highest score. *None* values in slices represent unused column in the slice.

average_error_: float

Mean value of the input error.

top_slices_statistics_: list of dict of length `len(top_slices_)`

The statistics of the slices found sorted by slice's scores. For each slice, the following statistics are stored:

- `slice_score`: the score of the slice (defined in `_score` method)
- `sum_slice_error`: the sum of all the errors in the slice
- `max_slice_error`: the maximum of all errors in the slice
- `slice_size`: the number of elements in the slice
- `slice_average_error`: the average error in the slice (`sum_slice_error / slice_size`)

5.1.3 References

SliceLine: Fast, Linear-Algebra-based Slice Finding for ML Model Debugging, from *Svetlana Sagadeeva* and *Matthias Boehm* of Graz University of Technology.

S

Slicefinder (*class in sliceline*), 11