Adaptive Neighbourhoods

Jay Morgan, Adeline Paiement, Arno Pauly, and Monika Seisenber

Oct 03, 2022

FIRST STEPS

This is the documentation for the Adaptive Neighbourhoods algorithm.

CHAPTER ONE

INSTALLATION GUIDE

To install…

CHAPTER

TWO

EXAMPLES

Some examples of using the algorithm

CHAPTER THREE

API

adaptive_neighbourhoods.neighbourhoods.**epsilon_expand**(*x: ~numpy.ndarray, y: ~numpy.ndarray, step_size: ~typing.Union[int, float] = 1e-07, distance_fn: ~typing.Callable[[~typing.Union[~numpy.ndarray, int, float], ~typing.Union[~numpy.ndarray, int, float]], ~typing.Union[~numpy.ndarray, int, float]] = <function inverse_quadric>, iterative: bool = False*) → ndarray

Create adapted and unique neighbourhoods.

Considering the density information and class distribution of *x* and *y*, generate the upper bound of the neighbourhoods to search for adversarial examples.

Parameters

- **x** (*np.ndarray*) – The input data to generate neighbourhoods around, these will generally be your training data for which adversarial neighbourhoods will be searched within.
- **y** (*np.ndarray*) – Encoded repreentations of classes for each data point in *x*. For example, if we have a two class problem with 4 data points, *y* could look like np.array([0, 1, 1, 0]).
- **step_size** (*Number*) – The initial step size for expanding the neighbourhoods. This number should be suitably small so as to not overlap with any other points from different classes before being modulated by the density information.
- **distance_fn** (*Callable[[Arr, Arr], Arr]*) – A callable metric function to determine the distance between neighbouring data points from which estimated density is computed. By default, we use the inverse multiquadric function.
- **iterative** (*bool*) – Boolean parameter that, if True, will return neighbourhood size from each successive iteration of the algorithm that allows you to visualise the expansion of the neighbourhoods.

Returns

A numpy array of neighbourhood sizes for each data point in *x*. If iterative is True, then the return will contain the neighbourhood sizes for each iteration of this algorithm.

Return type

np.ndarray

Examples

FIXME: Add docs.

CHAPTER FOUR

ALGORITHMIC PERFORMANCE

This iterative algorithm has been profiled…

CHAPTER FIVE

CONTRIBUTING

CHAPTER SIX

INDICES AND TABLES

- genindex
- modindex
- search

INDEX