
PyLOD Documentation

Versión 1.0

Miguel Ángel López Robles

28 de agosto de 2019

Contents:

1. Introduccion	1
2. LOF	3
3. LOOP	5
4. LDOF	7
5. PINN-LOF	9
6. OUTRES	11
7. ODIN	13
8. MeanDIST	15
9. KDIST	17
Índice	19

CAPÍTULO 1

Introduccion

Bienvenido a PyDBOD, la biblioteca de Python para la detección de anomalías usando algoritmos basados en distancias. En esta biblioteca tienes una amplia selección de algoritmos los cuales vamos a documentar a continuación. El uso de todos se reduce a la creación de un objeto de la clase respectiva y el uso del método **fit_predict**.

Para instalar el paquete o obtener una distribución usar el repositorio en github o en PyPI;

<https://github.com/miki97/TFG-OutlierDetection>

<https://pypi.org/project/PyDBOD/>

Local Outlier Factor (LOF), o en español factor de valor atípico local, es una cuantificación del valor atípico de un punto perteneciente al conjunto de datos. Esta cuantificación es capaz de ajustar las variaciones en las densidades locales.

LOF ($k = 20$)

Constructor para la creación del objeto de la clase LOF.

Parámetros `int` – k , número de k vecinos a calcular

Tipo del valor devuelto objeto de la clase LOF

fit_predict (*data*)

Método para aplicar el algoritmo LOF a una matriz de datos.

Parámetros `numpy.array` – *data*, matriz de datos

Tipo del valor devuelto `numpy.array` de puntuaciones de anomalía

Local Outlier Probability (LoOP), esta técnica combina varios conceptos. En primer lugar, la idea de localidad, los algoritmos basados en densidad como LOF. Por otro lado, LOCI con conceptos probabilísticos.

LOOP ($k = 20$, $lamda=3$)

Constructor para la creación del objeto de la clase LOOP.

Parámetros

- **int** – k , número de k vecinos a calcular
- **int** – $lamda$, párametro para regular la normalización

Tipo del valor devuelto objeto de la clase LOOP

fit_predict ($data$)

Método para aplicar el algoritmo LOOP a una matriz de datos.

Parámetros **numpy.array** – $data$, matriz de datos

Tipo del valor devuelto **numpy.array** de probabilidad anomalia [0-1]

Local Outlier Probability (LoOP), utiliza la distancia relativa de un objeto a sus vecinos para medir la cantidad de objetos que se desvían de su vecindario disperso.

LDOF ($k = 20$)

Constructor para la creación del objeto de la clase LDOF.

Parámetros `int` – k , número de k vecinos a calcular

Tipo del valor devuelto objeto de la clase LOOP

fit_predict (*data*)

Método para aplicar el algoritmo LDOF a una matriz de datos.

Parámetros `numpy.array` – *data*, matriz de datos

Tipo del valor devuelto `numpy.array` de puntuaciones de anomalía

Projection-Indexed Nearest-Neighbour (PINN), en este algoritmo se propone un método de detección de valores atípicos locales proyectivo basado en LOF.

PINN-LOF(**k** = 20, **t**=2, **s**=1, **h**=20)

Constructor para la creación del objeto de la clase PINN-LOF.

Parámetros

- **int** – k, número de k vecinos a calcular
- **int** – t, probabilidad de seleccion de características para la proyección
- **int** – s, probabilidad de selección para la proyección
- **int** – h, número de k vecinos a calcular en la proyección

Tipo del valor devuelto objeto de la clase PINN-LOF

fit_predict(*data*)

Método para aplicar el algoritmo PINN-LOF a una matriz de datos.

Parámetros **numpy.array** – data, matriz de datos

Tipo del valor devuelto **numpy.array** de puntuaciones de anomalía

OUTRES

Outres es un algoritmo que propone desarrollar una puntuación de anomalías basada en la desviación de objetos en las proyecciones subespaciales. Para la selección de dichos subespacios se analiza la uniformidad de los datos en ellos.

OUTRES (*epsilon=15, alpha=0.01*)

Constructor para la creación del objeto de la clase OUTRES.

Parámetros

- **int** – epsilon, radio para la selección del vecindario
- **float** – alpha, limite de uniformidad que se permite como interesante

Tipo del valor devuelto objeto de la clase OUTRES

fit_predict (*data*)

Método para aplicar el algoritmo OUTRES a una matriz de datos.

Parámetros **numpy.array** – data, matriz de datos

Tipo del valor devuelto **numpy.array** de puntuaciones de anomalía

Outlier Detection using Indegree Number (ODIN), es un algoritmo que hace uso del grafico de los k-vecinos más cercanos y usa el grado de los nodos para el calculo de anomalías

ODIN ($k=20$, $t=0.01$)

Constructor para la creación del objeto de la clase ODIN.

Parámetros

- **int** – k, número de k vecinos a calcular
- **int** – t, umbral de dicisión

Tipo del valor devuelto objeto de la clase ODIN

fit_predict (*data*)

Método para aplicar el algoritmo ODIN a una matriz de datos.

Parámetros **numpy.array** – data, matriz de datos

Tipo del valor devuelto **numpy.array** de decisión 1-0

MeanDIST

El algoritmo MeanDIST usa la la media de las distancias en su vecindario para ordenar a los vérticesy seleccionar los que más se desvian.

MeanDIST ($k=20$, $t=1.5$)

Constructor para la creación del objeto de la clase MeanDIST.

Parámetros

- **int** – k , número de k vecinos a calcular
- **int** – t , parámetro para ampliar o reducir el umbral.

Tipo del valor devuelto objeto de la clase MeanDIST

fit_predict ($data$)

Método para aplicar el algoritmo MeanDIST a una matriz de datos.

Parámetros **numpy.array** – $data$, matriz de datos

Tipo del valor devuelto **numpy.array** de decisión 1-0

KDIST

El algoritmo KDIST el máximo de las distancias a sus k-vecinos más cercanos para ordenar a los vértices y seleccionar los que más se desvían.

KDIST ($k=20, t=1.5$)

Constructor para la creación del objeto de la clase KDIST.

Parámetros

- **int** – k, número de k vecinos a calcular
- **int** – t, parámetro para ampliar o reducir el umbral.

Tipo del valor devuelto objeto de la clase KDIST

fit_predict (*data*)

Método para aplicar el algoritmo KDIST a una matriz de datos.

Parámetros **numpy.array** – data, matriz de datos

Tipo del valor devuelto **numpy.array** de decisión 1-0

- **genindex**
- **modindex**
- **search**

F

`fit_predict()` (*función incorporada*), 3, 5, 7, 9, 11,
13, 15, 17

K

`KDIST()` (*función incorporada*), 17

L

`LDOF()` (*función incorporada*), 7

`LOF()` (*función incorporada*), 3

`LOOP()` (*función incorporada*), 5

M

`MeanDIST()` (*función incorporada*), 15

O

`ODIN()` (*función incorporada*), 13

`OUTRES()` (*función incorporada*), 11