PDStoolkit User Manual

Version: 0.0.2
Released: August 1, 2023

Table of Contents

1.0 About PDStoolkit Package

PDStoolkit (Process Data Science toolkit) is a Python package that has modules designed to make development of data-science solutions for process systems engineering (PSE) faster and easier. The current focus is on providing modules that make it easy to build solutions for process monitoring and fault diagnosis. Modules catering to other PSE needs will be provided in upcoming releases. In the majority of the cases, the modules in PDStoolkit package extends the base Sklearn classes to provide the additional functionalities which enable easy development of the aforementioned tools.

Installation and Sample usage

The easiest way to install PDStoolkit is to use *pip* as shown below.

```
pip install PDStoolkit
```

After installation, the PDStoolkit modules can be imported into your script as follows

```
from PDStoolkit import PDS_PCA
```

The following sample code builds a PCA-based process monitoring model using PDS-PCA class and uses it for subsequent fault detection and fault diagnosis on test data.

```
# import
from PDStoolkit import PDS_PCA

# fit PDS_PCA model
pca = PDS_PCA()
pca.fit(data_train_normal, autoFindNLatents=True)

T2_train, SPE_train = pca.computeMetrics(data_train_normal, isTrainingData=True)
T2_CL, SPE_CL = pca.computeThresholds(method='statistical', alpha=0.01)
pca.draw_monitoring_charts(title='training data')
```

Available modules

PDStoolkit currently provides five modules. These are listed in the table below. The applications that can be built using them are also mentioned.

Module	Model	Targeted Application(s)
PDS_PCA	Principal Component Analysis (PCA)	Fault detection; Fault diagnosis
PDS_PLS	Partial Least Squares (PLS) Regression	Fault detection
PDS_DPCA	Dynamic Principal Component Analysis	Fault detection
PDS_DPLS	Dynamic Partial Least Squares Regression	Fault detection
PDS_CVA	Canonical Variate Analysis	Fault detection

Available methods

The modules of the package have several common methods. The task executed in these methods are listed below.

Method	Purpose
fit()	implements the respective algorithm
computeMetrics()	computes monitoring metrics such as Q, T^2 for PCA model
computeThresholds()	computes thresholds for the monitoring metrics using monitoring metrics computed for the training dataset
draw_monitoring_charts()	draws monitoring charts over training or test data for the computed monitoring metrics
detect_abnormalities()	detects abnormal observations by comparing the monitoring metrics against the respective thresholds
get_contributions()	[if available] provides the contribution plots for the computed monitoring metrics for the given sample

2.0 Modules

Some relevant details on each module are provided below.

PDS_PCA

This module implements the classical principal component analysis. This module can be used to create fault detection and diagnosis (FDD) solutions for linear, gaussian, and static (where observations are not temporally correlated) processes.

Hyperparameters:

The number of retained principal components is the major hyperparameter. This value can be explicitly specified during model object's creation, or the module could be directed to compute it automatically: cumulative percentage variance technique is utilized for this.

PDS_PLS

This module implements the classical partial least squares regression. This module can be used to create fault detection solutions for linear, gaussian, and static (where observations are not temporally correlated) processes.

Hyperparameters:

The number of retained latent variables is the major hyperparameter. This value can be explicitly specified during model object's creation, or the module could be directed to compute it automatically: k-fold cross-validation technique is utilized for this.

PDS_DPCA

This module implements the dynamic principal component analysis. This module can be used to create fault detection solutions for linear, gaussian, and dynamic processes.

Hyperparameters:

The number of lags and the number of retained principal components are the major hyperparameters. While the number of lags needs to be explicitly specified (in the current version of PDStoolkit) during model object's creation, the number of principal components could be auto estimated via the cumulative percentage variance technique.

PDS_DPLS

This module implements the dynamic partial least squares regression. This module can be used to create fault detection solutions for linear, gaussian, and dynamic processes.

Hyperparameters:

The number of lags and the number of retained latent variables are the major hyperparameters. While the number of lags needs to be explicitly specified (in the current version of PDStoolkit) during model object's creation, the number of principal components could be auto estimated via the cumulative percentage variance technique.

PDS_CVA

This module implements the classical canonical variate analysis. This module can be used to create fault detection solutions for linear, gaussian, and dynamic processes.

Hyperparameters:

The lag order and the model order are the major hyperparameters. While the lag order needs to be explicitly specified (in the current version of PDStoolkit) during model object's creation, the model order could be auto estimated: the Hankel singular values are used for this. Model order, *n*, is chosen such that the normalized values of the $(n+1)^{th}$ onwards singular values are below the *ratioThreshold* parameter.

3.0 Resources

The following resources have been provided to help readers familiarize themselves with the package

	Purpose	Link
Module docs	Details on method parameters and method outputs for each module	Link
Tutorials	Show applications of PDStoolkit for • fault detection with Tennessee Eastman dataset • fault detection and diagnosis for a polymer manufacturing facility • fault detection a polyethylene manufacturing process	Link

Interested readers can check out our books (https://mlforpse.com/books/) which cover the algorithms available in PDStoolkit and several other ML algorithms (that have proven useful in process industry) in details. In these books, step-by-step instructions, supported with real process datasets, show how to develop ML-based solutions for process monitoring, predictive maintenance, fault diagnosis, soft sensing, and process control.

---------- End of Manual ----------

4.0 Revision History

Version No.	Revision Date	Changes/Notes
V0.0.1	Aug 1, 2023	Initial Version