MASE

Release 0.0.7

Marco Christiani

Dec 16, 2020

Table of Contents

1 Installation **1**

2 Overview **3**

3 Simulating New Data **5**
3.1 Create a Simulation Object 5

Index **9**

Installation

At the command line:

```
pip3 install mase
```

From a Jupyter Notebook cell:

```
!pip3 install mase
```

Overview

class mase.**Simulation** (*n_observations*, *means: numpy.ndarray = None*, *covariance_matrix: numpy.ndarray = None*)

A Simulation object stores a Pandas DataFrame where the number of features is determined by the means or covariance_matrix arguments supplied at initialization. If neither means nor covariance_matrix is supplied, the number of features will be set to n_observations thus the DataFrame will be square.

Currently, MASE only supports simulation of multivariate normal data.

Parameters

- **n_observations** – number of observations to simulate
- **means** – Optional; numpy array of means corresponding to each feature
- **covariance_matrix** – Optional; numpy array of covariance matrix that you would like the simulated data to emulate

add_gaussian_observations (*summary_df*, *feature_index*, *df=None*, *visualize=False*, *append=False*)

Parameters

get_data ( )

Getter for DataFrame of Simulation object

Returns Pandas DataFrame

Chapter 3

Simulating New Data

3.1 Create a Simulation Object

Getting started with MASE is straightforward, let's look at an example.

Covariance Matrix: Let's simulate 5 indepedent features by setting this to the 5x5 identity matrix `I\in \mathbb{R}^{5\text{x}5}`

Means: Let's choose each feature to have mean 0 by not supplying a `means` argument

N: Let's generate 100 observations.

```
    cov = np.eye(5)  # 5 independent features all with 0 mean
    sim = Simulation(100, covariance_matrix=cov)  # 100 observations
```

Great! Now we have a `Simulation` object created and we can begin adding anomaly patterns.

First, let's decide what anomalous behavior we would like to add to the data and store that information in a Pandas DataFrame called `summary_df`

```
    summary_df = pd.DataFrame()
    summary_df['mean'] = [3, 0]
    summary_df['sd'] = [1, 2]
    summary_df['n_obs'] = [20, 10]
```

summary_df now looks something like this:

mean	sd	n_obs
3	1	20
0	2	10

This dataframe corresponds to the adding of:

- `20` observations `N \sim (3, \sigma)`
- `10` observations `N \sim (0, 2\sigma)`

Let's apply this to feature 0 in our data:

```
    feature_index = 0
    sim.add_gaussian_observations(summary_df, feature_index, visualize=True)
```

Go ahead and give this code a try yourself by running .. code-block:: python

mase.basic_demo()

Which runs the following function: .. literalinclude:: ../../mase/demo.py

lines 4-14

- *Index*
- *Module Index*
- *Search Page*

A

add_gaussian_observations() (mase.Simulation method), 3

G

get_data() (mase.Simulation method), 3

S

Simulation (class in mase), 3

A

add_gaussian_observations() (mase.Simulation method), 3

G

get_data() (mase.Simulation method), 3

S

Simulation (class in mase), 3