Data Approximation using Kriging

Group 12

Suraj Sanka	Mrinalgouda Patil	Vinod Metla
130010057	130010061	130010048

SDES Project 2
21st November, 2016

Contents

1. Problem Statement
2. GitHub page
3. GitHub
4. Commits
5. Tests
6. Automation
7. Documentation
8. Code Cleanliness
9. Using the program

Problem Statement

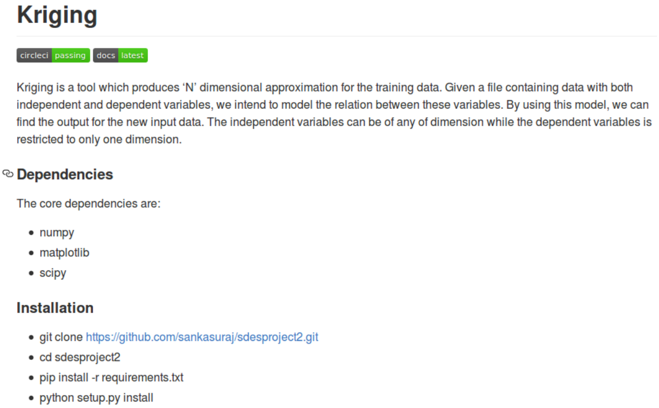
To create a tool which produces approximation for any given ‘N’ dimensional training data.

Given a file containing data with both independent and dependent variables, we intend to model the relation between these variables. By using this model, we can find the output for the new input data.

Note: The independent variables can be of any of dimension while the dependent variables is restricted to only one dimension.

GitHub page

Visit out GitHub page at
`https://github.com/sankasuraj/sdesproject2`



Kriging

Kriging is a tool which produces 'N' dimensional approximation for the training data. Given a file containing data with both independent and dependent variables, we intend to model the relation between these variables. By using this model, we can find the output for the new input data. The independent variables can be of any of dimension while the dependent variables is restricted to only one dimension.

Dependencies

The core dependencies are:

- numpy
- matplotlib
- scipy

Installation

- git clone https://github.com/sankasuraj/sdesproject2.git
- cd sdesproject2
- pip install -r requirements.txt
- python setup.py install

Figure : GitHub page

Commits and branches

Total commits made till submission: 94

1. Mrinal Patil - 52
2. Suraj Sanka - 24
3. Vinod Kumar - 18

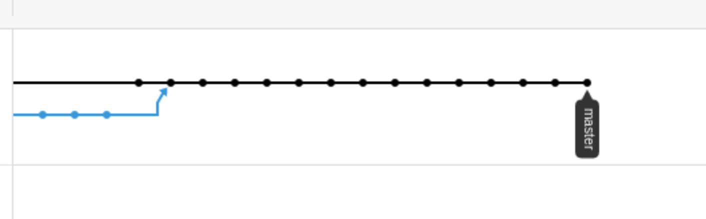


Figure : GitHub branches

Git Commits

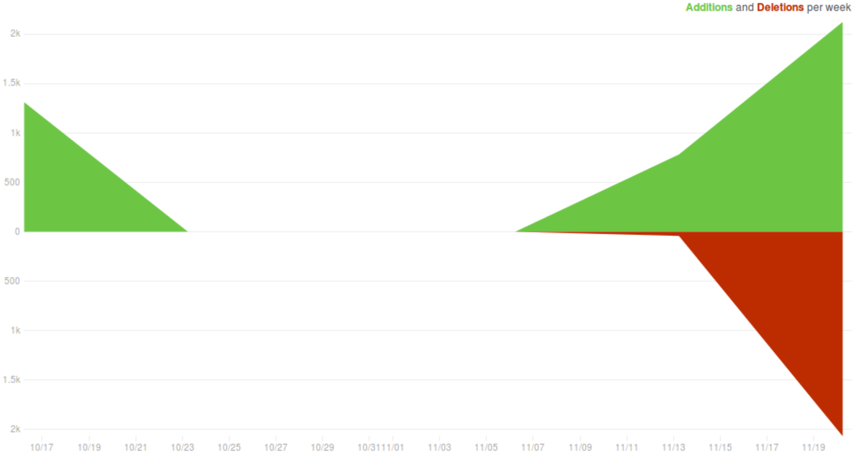
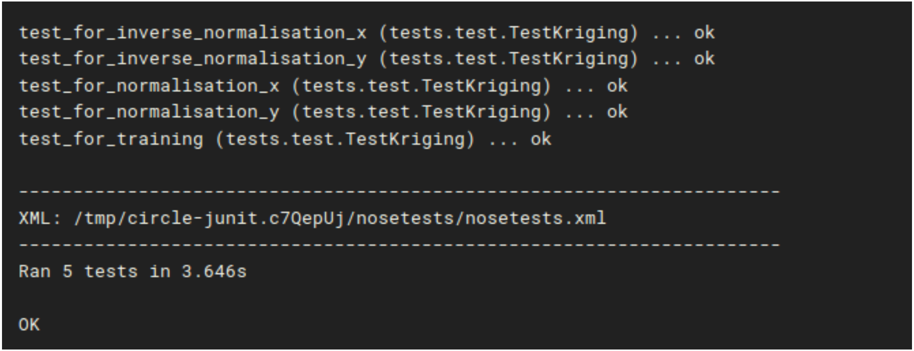


Figure : Git Commits timeline

Tests

1. Nose, Pytest, Unittest used
2. circle-ci testsuites in verbose mode
3. Click on circleci in the github page to run the tests
4. Alternatively run **make test** to test the code
5. mock is not used

Verbose testsuite



```
test_for_inverse_normalisation_x (tests.test.TestKriging) ... ok
test_for_inverse_normalisation_y (tests.test.TestKriging) ... ok
test_for_normalisation_x (tests.test.TestKriging) ... ok
test_for_normalisation_y (tests.test.TestKriging) ... ok
test_for_training (tests.test.TestKriging) ... ok

----------------------------------------------------------------------
XML: /tmp/circle-junit.c7QepUj/nosetests/nosetests.xml
----------------------------------------------------------------------
Ran 5 tests in 3.646s

OK
```

Figure : On clicking circleci in GitHub page

Automation

1. make file
   1. **make** command will start the program
   2. **make test** will run all the tests
2. setup.py - yes
3. circle-ci is used not travis-ci

Documentation

Docs available at `http://kriging.readthedocs.io/`

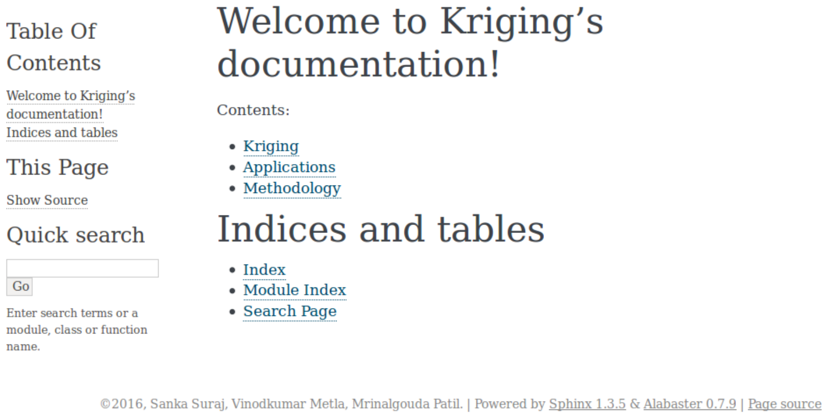


Table Of Contents

- Welcome to Kriging’s documentation!
- Indices and tables

This Page

- Show Source

Quick search

Go

Enter search terms or a module, class or function name.

Welcome to Kriging’s documentation!

Contents:

- Kriging
- Applications
- Methodology

Indices and tables

- Index
- Module Index
- Search Page

©2016, Sanka Suraj, Vinodkumar Metla, Mrinalgouda Patil. | Powered by Sphinx 1.3.5 & Alabaster 0.7.9 | Page source

Figure : Documentation

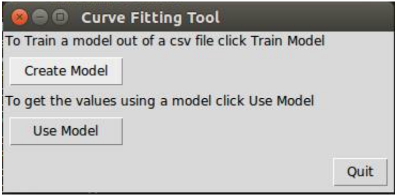
Code Cleanliness

PEP8 is followed

Using the program:

1. Making the model:
    1. Make
    2. Choose the csv file which has x, y data
    3. Choose a name for the model to save
    4. Wait until the model is trained
2. Using the model:
    1. Make
    2. Choose a previously saved model
    3. Choose an x data where y is to be found
    4. Choose a name for the output file
    5. Wait until the model finds the value for each row in x data

Using the program

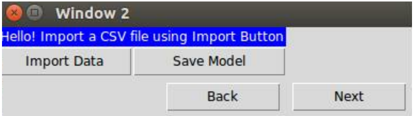


Curve Fitting Tool

To Train a model out of a csv file click Train Model

To get the values using a model click Use Model

Figure : First block

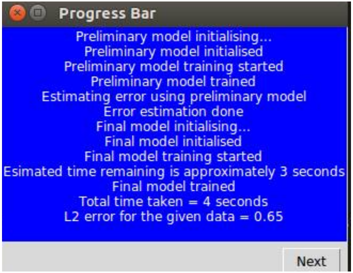


Window 2

Hello! Import a CSV file using Import Button

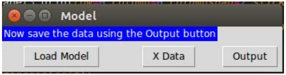
Figure : Second block

Using the program



```
Preliminary model initialising...
Preliminary model initialised
Preliminary model training started
Preliminary model trained
Estimating error using preliminary model
Error estimation done
Final model initialising...
Final model initialised
Final model training started
Esimated time remaining is approximately 3 seconds
Final model trained
Total time taken = 4 seconds
L2 error for the given data = 0.65
```

Figure : Progress window



Now save the data using the Output button

Figure : Using the model

Bibliography

- Kriging.
  pykriging.com
  https://github.com/capaulson/pyKriging
- Tkinter
  https://www.summet.com/dmsi/html/guiProgramming.html
- Circle Ci. https://circleci.com/docs/getting-started/

Thank you