**Documentation of Synthetic Data (Interact) for project JiaXing**

**Preamble**

The data synthesised/transformed from real projects are designed for testing the effectiveness and efficiency of each of the JiaXing tuning projects (YangZhou etc).

They are stored in .json file format, with ‘num\_arg\_vals’ being a list that captures the number of values in each dimension (and also how many dimensions are in this synthetic data), ‘synthetic data’ storing the synthesised randomly generated tuning scores, ‘theoretical data’ storing the synthesised mean of randomly generated tuning scores. The final object in the json is a dictionary under ‘max’, which contains the synthesised maximum score ‘synth\_max’, theoretical (mean of randomly generated) maximum score ‘theor\_max’, synthesised maximum coordinates ‘synth\_max\_coord’ and theoretical maximum coordinates ‘theo\_max\_coord’.

All data also come with a csv storing the data in a tabular format (X features = coordinates, y = score) for the generated datapoints, which could be used to fit a gaussian process regressor on the fly for tests that require continuous values

**Documentation**

This dataset contains 3 batch of data

Batch 1: Synthesised data with each type of characteristics captured (576; 500 mb)

Batch 2: Synthesised data with randomly chosen characteristics (120; 101 mb)

Batch 3: Real accuracy/r2 data from past projects (50; 5.1MB)

Batch 1 2 and 3

General concepts:

Each point in the synthetic data is generated by random generator according to:

1. Get the sum of the means of each coordinates, as well as the added mean from two way interactions
2. The sum of the vector is added to 0.5 to gain this observation

\*note observations may go above 1 or subzero

**Mu** (mean vector)

For each dimension (each individual hyperparameter), it is assumed with rising values of actual hyperparameter values, the ‘mean’ of the contribution to the score.

Each dimension thus can be generated with a trend/pattern:

‘take\_off’: first few values contribute 0, then gradually increasing by 0.01

‘flat’: all values contribute 0

‘v\_shape’: values first decrease, then increase

‘down\_flat\_up’: values first decrease, then contribute 0, then increase again

‘up\_flat\_up’: values first increase, then contribute 0, then increase again

‘flat\_up\_flat’: values first contribute 0, then increase, then contribute 0

‘flat\_up\_down\_flat’: values first contribute 0, then increase, then decrease, then contribute 0

\*an error existed so ‘flat\_up\_down\_flat’ was not included in final simulations

Note: all steps in trends move in magnitude of 0.01. (either -0.01, 0 or +0.01)

Line chart

Description automatically generated with medium confidenceNote: when in action, all these points of change are randomly generated, within a reasonable space (i.e. won’t take\_off from the leftmost point)

**sd**

The standard deviations are

0.005

**Interaction**

-interaction between pairs can either be

-0.0001, 0, 0.0001

Batch 1:

D\_VAL = {2: (5, 7, 'rand'),

3: (5, 7, 'rand'),

4: (5, 7, 'rand'),

5: (5, 7, 'rand')

}

for d in (2, 3, 4, 5):

for d\_val in D\_VAL[d]:

for pattern in ['take\_off', 'v\_shape', 'down\_flat\_up', 'up\_flat\_up', 'flat\_up\_flat', 'rand']:

if pattern == 'flat\_up\_down\_flat' and (d\_val == 5 or d\_val == 'rand'):

continue

for sd in (0.005,):

for interaction in (-0.0001, 0, 0.0001, 'rand'):

for rep in (1, 2):

Note: each combination repeated two times

Note: when d\_val = 5, or there is chance of <=7 in d\_val = ‘rand’, will skip ‘flat\_up\_down\_flat’ as hyperparameter mean pattern as the generating mechanism of ‘flat\_up\_down\_flat’ requires there to be at least 7 values in the dimension

Note: when pattern is not rand, all dimensions utilise the same pattern

Note: when d\_val is not rand, all dimensions have the same values

Note: when interaction is not rand, all dimensions have the same two-way-interaction

Batch 2:

D\_VAL = {2: (5, 7, 'rand'),

3: (5, 7, 'rand'),

4: (5, 7, 'rand'),

5: (5, 7, 'rand'),}

for d in (2, 3, 4, 5):

for d\_val in D\_VAL[d]:

for pattern in ['rand']:

if pattern == 'flat\_up\_down\_flat' and (d\_val == 5 or d\_val == 'rand'):

continue

for sd in (0.005,):

for interaction in ('rand',):

for rep in range(10):

Note: each combination repeated ten times

Note: all other notes same as batch 1

Batch 3

-comes from projects including ML Asmt 2; ADS Asmt 1; ADS Asmt 2 and New Brownlow Predictor project

-converted into same format as synthesised data using *‘./Batch 4 5 6 Creator/synthetic\_data\_generator\_batch4.ipynb’*

Batch 3

-accuracy, r2