Heterogeneous software effort estimation readme

This is a heterogeneous software effort estimation project.

# Requirements

* numpy (1.13.3)
* chainer (3.1.0)
* matplotlib (2.1.0)
* pandas (0.20.3)

# Description of the folders or files

Description of the items of the project are tabulated in Table 1.

Table 1 description of the items in the project

|  |  |
| --- | --- |
| **File or directory name** | **Description** |
| data | Temporary folder for intermediate files. |
| datasets | All the datasets saved in this directory. |
| models | All the trained models saved in this directory. |
| read\_data\_validation.py | Script for reading data. |
| criteria.py | Script for evaluation measures. |
| Within\_company.py | Script for the within-company SEE. |
| HSEE\_static\_scenario.py | Script for the static HSEE scenario. |
| HSEE\_dynamic\_scenario.py | Script for the dynamic HSEE scenario. |

# How to run this code

In our approach, we category HSEE into two cases: static HSEE and dynamic HSEE.

Static HSEE:

Readers can run the **HSEE\_static\_scenario.py** to validate the accuracy of the static HSEE estimator.

In HSEE\_static\_scenario.py file, readers can modify the source dataset and the target dataset of the main function to validate the performance of different ‘source-target’ pairs. For example, we want to set the ‘cocnas’ as the source and ‘opens’ as the target, parameter “**data\_set\_name=['cocnas', 'opens']”** of the main function can be settled.

When the training procedure is finished, the result of static HSEE will be saved into the model (prevModel.csv under the data folder). The prevModel.csv file will be used in the dynamic HSEE, when there existing other source datasets.

Dynamic HSEE:

Since training a dynamic HSEE model is based on an incremental strategy, we only need to add datasets into the pre-existing model continually to simulate the dynamic scenario. For example, assuming dataset ‘China’ is used as target, the ‘Cocas’ and ‘opens’ are the source datasets (‘Cocas’ comes early than ‘opens’). To simulate the dynamic HSEE scenario, we firstly put the ‘Cocas’ and ‘China’ datasets into the **HSEE\_static\_scenario.py** model and **get the** **initialized model** ‘Cocas-China’. Then we add the ‘opens’ dataset into the obtained model ‘Cocas->China’ and get the incremental model ‘[opens, Cocas]->China’. It is noteworthy that readers must run the **HSEE\_dynamic\_scenario.py** file to **increase the newly arrived** ‘opens’ into the pre-obtained model ‘Cocas->China’ (not the HSEE\_static\_scenario.py).

In **HSEE\_dynamic\_scenario.py**, parameter “**data\_set\_name=['prevModel', 'Newly\_dataset']**” is used to set **the pre-obtained model and the newly arrived source dataset**. For example, the “data\_set\_name=['prevModel', 'China']” indicated that reading pre-obtained model from ‘prevModel.csv’ and putting the ‘China’ dataset as the newly arrived dataset into the model to be trained. Parameter ‘**save\_code**’ is used to indicate the status of current task. For example, the ‘save\_code=True’ represents that the newly arrived dataset is helpful to improvement the accuracy of the dynamic HSEE model, so we save the obtained model. The ‘save\_code=false’ is only used to check whether the newly arrived dataset is beneficial to improve the accuracy of the dynamic HSEE model

There are several parameters you should be pay attention, such as data\_set\_name, batch\_size and validation\_patience\_original.

There are some feasible parameters：

|  |  |  |
| --- | --- | --- |
| data\_set | batchsize | validation\_patience\_original |
| china | 50 | 30 |
| kitchenham | 15 | 20 |
| albrecht | 5 | 300 |
| maxwell | 15 | 30 |
| cocnas | 15 | 40 |
| opens | 15 | 30 |
| kemerer | 10 | 5 |
| miyazaki94 | 15 | 200 |