Clustering-Based Ground Motion Selection Example

Last updated on 07_01_2024 by Yiming Jia

Notes:

1. The “Example” folder includes the codes and data to reproduce the ground motion (GM) selection presented in Section 4.1 of Jia and Sasani (2024).

2. The results of GM pre-selection and the ones obtained by knowledge of earthquake engineering are included. The corresponding steps are marked as “finished”, which indicates no need to run.

3. Considering that some codes may take several hours to run, the corresponding steps are marked as “time-consuming”.

4. Considering that random sampling is used for different purposes (e.g., generating pre-trained GMs, weight initialization of autoencoder, realizations of GM response spectra, etc.), reproducing this example may not always result in the exact same GMs presented in Section 4.1 of Jia and Sasani (2024). However, the selected GMs can always match the conditional spectra’s mean and variability well while fully describing the candidate GMs.

Pre-request:

The user needs to have access to Python 3.10.14 (Tensorflow 2.13.1 required) and Matlab R2023b.

The authors recommend running Python codes on Jupyter Notebook. Running the codes on different versions of Python, Tensorflow, or Matlab may result in compatibility issues. However, most potential compatibility issues can be easily resolved by following the suggestions in the error messages.

Step-by-step instruction:

Step 1 (finished): Pre-select pulse-type and non-pulse-type GMs based on knowledge of earthquake engineering to obtain candidate GMs for a structure with a fundamental period of 1.0s in San Francisco, California.

Step 2 (finished): Calculate and scale the logarithmic response spectra (lnSa) of candidate GMs and save them as a matrix as MATLAB data “GMIM_Pulse.mat” and “GMIM_NonPulse.mat” .

Step 3 (time-consuming): Run MATLAB codes "Pretrain_Generation_Pulse.m" and "Pretrain_Generation_NonPulse.m" to generate the pretrain data.

This step may take a couple of hours, depending on the number of pre-trained GMs and the available computational resources.

The generated data are saved as "GMIM_Pulse_Pretrain.mat" and "GMIM_NonPulse_Pretrain.mat". Considering this step is time-consuming, the results ("GMIM_Pulse_Pretrain.mat" and "GMIM_NonPulse_Pretrain.mat") are provided.

Step 4 (time-consuming): Run Python codes "Autoencoder_Pulse_Pretrain.ipynb" and "Autoencoder_NonPulse_Pretrain.ipynb" on Jupyter Notebook.

This step may take several hours, depending on the number of candidate GMs and the available computational resources.

Each code automatically generates three folders. For the pulse-type GM case, these generated folders are

a) Data_Pulse_Pretrain (includes the latent features, reconstructed lnSa, loss, and running time),

b) Figure_Pulse_Pretrain (includes CAE architecture, quantile-quantile plot of lnSa, and loss plot),

c) Model_Pulse_Pretrain (includes the checkpoint, which is used in the following fine-tuning process).

Considering this step is time-consuming, the above-mentioned three folders are provided. Note that running the Python codes can overwrite the files in the above-mentioned three folders.

Step 5: Run Python codes "Autoencoder_Pulse_Finetune.ipynb" and "Autoencoder_NonPulse_Finetune.ipynb" on Jupyter Notebook.

For each code, three folders are automatically generated (similar to Step 4).

The above-mentioned three folders are provided. Note that running the Python codes can overwrite the files in the above-mentioned three folders.

Step 6: Run MATLAB codes "Kmeans_Pulse.m" and "Kmeans_NonPulse.m".

The results are saved as Matlab data "Cluster_Pulse.mat" and "Cluster_NonPulse.mat".

Step 7 (finished): Calculate the ground motion prediction equations using “GMPEs.xlsm” and knowledge of earthquake engineering, and save them as matrix as MATLAB data “GMPE.mat”.

Step 8 (time-consuming): Run MATLAB code “Generation_GM_Selection.m” to select the pulse-type and non-pulse-type GMs.

Note that this step may take a couple of hours, depending on the number of clusters, the number of realizations, and the available computational resources.

The output file “GM_Selection. mat” includes

a) ID_Pulse_Final and ID_NonPulse_Final (the ID for the selected pulse-type and non-pulse-type GMs as two vectors),

b) LogSa_Final (the lnSa of selected GMs as a matrix),

c) Mean_LogSa_Final (mean of lnSa of selected GMs as a vector),

d) Std_LogSa_Final (standard deviation of lnSa of selected GMs as a vector).

e) CMS (conditional mean spectra as a vector),

f) CMS_sigma (sigma of conditional mean spectra as a vector),

g) T (range of periods as a vector).

Considering this step is time-consuming, the output file “GM_Selection. mat” is provided. Note that running the Matlab code can overwrite the output file.

Step 9: Run MATLAB code “Plot_Selected_GMs.m” to plot the conditional spectra and the lnSa of selected GMs (for visualization purposes only).

Reference:

Jia, Y., and Sasani, M. (2024). “Convolutional Autoencoder-Based Ground Motion Clustering and Selection”, **In Review** (will be available after this paper is published).