CEE 154/254 Data Analytics for Physical Systems
Autumn 2023
Project/Competition – Test and submission instruction

Training and test files

Please download and unzip the file ‘*test_phase.zip*’. There are 27 data files in this archive. Each data file contains one or multiple MATLAB table matrices. Followings are descriptions of these files.

- **Short-term prediction:**
  - ‘*train_data_short_term_0_var.mat*’ and ‘*test_data_short_term_0_var.mat*’ are training and test data for short term prediction without adding noise.
  - ‘*train_data_short_term_5_var.mat*’ and ‘*test_data_short_term_5_var.mat*’ are training and test data for short term prediction after adding zero-mean Gaussian noise with 0.5 times the standard deviation of training PM 2.5 measurements.
  - ‘*train_data_short_term_10_var.mat*’ and ‘*test_data_short_term_10_var.mat*’ are training and test data for short term prediction after adding zero-mean Gaussian noise with 1.0 times the standard deviation of training PM 2.5 measurements.
- **Long-term prediction**:
  - ‘*train_data_long_term_0_var.mat*’ and ‘*test_data_long_term_0_var.mat*’ are training and test data for long term prediction without adding noise.
  - ‘*train_data_long_term_5_var.mat*’ and ‘*test_data_long_term_5_var.mat*’ are training and test data for long term prediction after adding zero-mean Gaussian noise with 0.5 times the standard deviation of training PM 2.5 measurements.
  - ‘*train_data_long_term_10_var.mat*’ and ‘*test_data_long_term_10_var.mat*’ are training and test data for long term prediction after adding zero-mean Gaussian noise with 1.0 times the standard deviation of training PM 2.5 measurements.
- **Interpolation**:
  - ‘*train_data_interpolation_0_var.mat*’ and ‘*test_data_interpolation_0_var.mat*’ are training and test data for interpolation without adding noise.
  - ‘*train_data_interpolation_5_var.mat*’ and ‘*test_data_interpolation_5_var.mat*’ are training and test data for interpolation after adding zero-mean Gaussian noise with 0.5 times the standard deviation of training PM 2.5 measurements.
  - ‘*train_data_interpolation_10_var.mat*’ and ‘*test_data_interpolation_10_var.mat*’ are training and test data for interpolation after adding zero-mean Gaussian noise with 1.0 times the standard deviation of training PM 2.5 measurements.

We also provide ‘*separated_train_data_*.mat*’ files containing the same data as the corresponding ‘*train_data_*.mat*’ files, but having static and mobile devices’ data separated into two table matrices. You can use these data files if you have different algorithms for mobile devices and static devices.

Testing metric

We will use the normalized root mean squared error (NRMSE) for evaluation. The project data description states that we will use the standard deviation of ground truth as the normalizer for NRMSE. However, to be consistent with the definition in Lecture 12, we will use the squared root normalizer:

$$NRMSE = \frac{RMSE(Prediction, ground\ truth)}{\sqrt{\sum(ground\ truth)^2}}$$

Note that using different normalizers will only change the scale of the resulting metrics.

Submission instruction

- Please save your M-by-1 prediction vector, ‘`pred_pm2d5`’, for each problem using the following matlab command:

  ```
  save([problem_name,'_',num2str(var_level),'.mat'],'pred_pm2d5'
  );
  ```

  where `problem_name` is each element from {‘long_term’, ‘short_term’, ‘interpolation’} for the three prediction types, respectively; and `var_level` is each element from {0, 5, 10] for the three noise standard deviation levels (i.e., 0, 0.5, 1.0), respectively.

- Add all 9 mat files and all your codes to archive ‘*LastName1_LastName2.zip*’, naming by last names of your team members. For example, ‘*Noh_Liu.zip*’.
- Then, submit your zip file to canvas.

Example code

The following is an example for your implementation:

```
problem_name = 'long_term';
problem_type = 2;
var_level = 0;
train_data = load(['train_data_',problem_name,'_',...
                num2str(var_level),'_var.mat']).train_data;
test_data = load(['test_data_',problem_name,'_',...
                num2str(var_level),'_var.mat']).test_data;

pred_pm2d5 = pm2d5_pred_model(train_data, test_data,
problem_type)

save([problem_name,'_',num2str(var_level),'.mat'],'pred_pm2d5');
```