Mape_Maker Documentation

Release 0.9

David L. Woodruff, Guillaume Goujard, Yifei Chen, Sukhmandeep

Dec 20, 2021

Contents

This package can be used to simulate scenarios of wind power forecasts from actuals or wind power actuals from forecasts. It has been implemented so that a generalization to any type of dataframe providing actuals and forecasts, is possible.

CHAPTER 1

Introduction

This package can be used to simulate scenarios of wind power forecasts from actuals and vice-versa. It has been implemented so that a generalization to any source of uncertainty providing actuals and forecasts, is possible.

The main inputs of the package are :

- an input dataset giving forecasts and actuals for specified datetimes as a csv file (see more at *Input*).
- a simulation input dataset (sid) giving at least one of the two columns (forecasts or actuals) for specified datetimes as a csv file. It can also be a subset of the input dataset, specified by a start and end-date. By default, the sid is the input dataset (see more at *Options*).
- r_tilde : a desired MAPE (i.e mean absolute percent error see more at *Percent Errors and MAPEs*) for the simulations in output
- user-specified technical parameters (see more at *Options*).

The mape_maker class estimates the conditioned distribution of the errors considering the input values. It adjusts these distribution to satisfy the specified target MAPE. Having fitted a base process, it simulates highly auto-correlated errors and finally if the user specifies it, it operates a curvature optimization. This approach will yield “plausible” scenario sets (see more at Plausibility objectives).

1.1 Percent Errors and MAPEs

We denote f and a as respectively the timeseries of forecasts and actuals. From there we can define two MAPEs depending on the simulation you wish to accomplish.

Then if you are simulating **forecasts from actuals**,

$$mape = \frac{100}{n} \sum_{i=1}^{n} \frac{|f_i - a_i|}{a_i}$$

If you are simulating **actuals from forecasts**,

$$mape = \frac{100}{n} \sum_{i=1}^{n} \frac{|f_i - a_i|}{f_i}$$

1.2 Plausibility objectives

A scenario set is said to be "plausible" if :

- the shape of the distribution of the scenarios errors is close to the shape of the empirical distribution of errors;
- the computed auto-correlation coefficients for the set are close the empirical values;
- the computed curvature for the set is close to the empirical value, especially when the scenarios are forecasts.

1.3 Setup

The package is compatible with Python version 3 or higher only. The user is expected to have installed pyomo before running the package. Go to http://www.pyomo.org for more information.

1. Switch to a proper directory and then type:

```
git clone + https://github.com/mape-maker/mape-maker.git
```

2. Install the package with the setup.py file:

```
python setup.py develop
```

3. For a quick-first run :

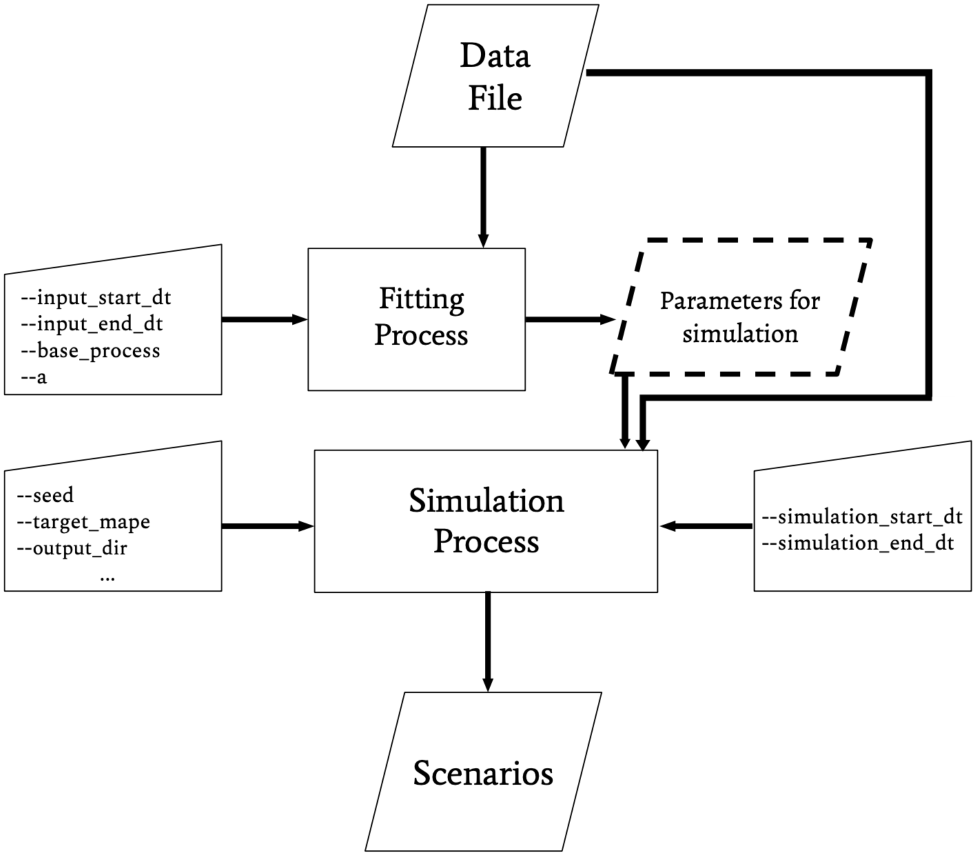
```
python -m mape_maker "mape_maker/samples/wind_total_forecast_actual_070113_063015.csv"
```

- If you want to optimize the curvature, you need to install a Quadratic MIP solver such as "Gurobi": http://www.gurobi.com (e.g Gurobi, Cplex, SCJP, etc.)

1.4 Notes about the Input File

1. If you want to use your own datafile as an input to run the mape_maker, then the input file format should be:
   - **"datetime"** as the first column, formatted as 'Y-M-D H:M:S'. i.e: 2020-01-01 01:00:00
   - **"forecasts"** data as the second column, format as "float". i.e: 3264.59
   - **"actuals"** data as the third column, format as "float". i.e: 3264.59
2. If the forecasts and actuals throughout the dataset are the same numbers up to most of the decimal points, then the software will not run the scenarios. This is the case because then there is little to no relative error, which leads to invalid values for r_tilde.
3. If the input datafile has any missing values, then the program will terminate with a KeyError warning.

1.5 Summary of the Algorithm


Data
File
--input_start_dt
--input_end_dt
--base_process
--a
Fitting
Process
Parameters for
simulation
--seed
--target_mape
--output_dir
...
Simulation
Process
--simulation_start_dt
--simulation_end_dt
Scenarios

CHAPTER 2

Citing

“Mape_Maker: A Scenario Creator” Guillaume Goujard, Jean-Paul Watson, and David L. Woodruff, Engergy Systems <http://link.springer.com/article/10.1007/s12667-020-00408-6>‘ 2020.

“Constructing probabilistic scenarios for wide-area solar power generation” David L. Woodruff, Julio Deride, Andrea Staid, Jean-Paul Watson, Gerrit Slevogt, César Silva-Monroy, Solar Engergy <https://doi.org/10.1016/j.solener.2017.11.067>‘ 2018.

CHAPTER 3

Input

The input of the package is assumed to be a csv giving forecasts and actuals for specified datetimes as a csv file.

3.1 Notes about the Input File

1. If you want to use your own datafile as an input to run the mape_maker, then the input file format should be:
    - **"datetime"** as the first column, formatted as 'Y-M-D H:M:S'. i.e: 2020-01-01 01:00:00
    - **"forecasts"** data as the second column, format as "float". i.e: 3264.59
    - **"actuals"** data as the third column, format as "float". i.e: 3264.59
2. If the forecasts and actuals throughout the dataset are the same numbers up to most of the decimal points, then the software will not run the scenarios. This is the case because then there is little to no relative error, which leads to invalid values for r_tilde.
3. If the input datafile has any missing values, then the program will terminate with a KeyError warning.

3.2 Example

We give an example as the first 10 rows of the csv located under mape_maker/samples/wind_total_forecast_actual_070113_063015.csv.

datetimes	forecasts	actuals
7/1/13 0:00	2031.94	1947.52095
7/1/13 1:00	1969.84	2074.72335
7/1/13 2:00	1902.99	2246.44718
7/1/13 3:00	1768.13	1978.91344
7/1/13 4:00	1708.09	1767.39892
7/1/13 5:00	1656.86	1635.56253
7/1/13 6:00	1410.82	1160.40714
7/1/13 7:00	966.72	489.04769
7/1/13 8:00	665.22	224.73994
7/1/13 9:00	406.82	196.72952

CHAPTER 4

Options

The options of the package are :

```
python mape_maker --help
```

4.1 Options with More Details

- **–input_sid_file TEXT**:

  The path to a simulation input dataset (sid) with one or two timeseries (e.g. actuals), from which scenarios for the other timeseries are generated (e.g. forecasts)

  The following loads "sid.csv" located under the current directory :

  ```
  --input_sid_file "sid.csv"
  ```

  ```
  -sf "sid.csv"
  ```

  If this option is not given, the sid will be taken as a subset of the input dataset, specified by a simulation_start_dt and simulation_end_dt.

- **–output_dir TEXT**:

Path to destination dir where the scenario are saved as csv file(s).

The following are the two ways to specify that the output directory is called "output":

```
--output_dir "output"
```

```
-o "output"
```

If this option is not given, the output directory is assumed to be None. No output directory will be created.

Note: If the output directory is not given, then the only output will be a png image of the plot showing the scenarios and saved under the current directory.

Warning: If the output directory already exists, the program will terminate and issue messages. It won't overwrite an existing directory.

- **–verbosity_output TEXT**:

  The name of the verbosity output file

  The following are two ways to specify the verbosity level:

  ```
  --verbosity_output "output.log"
  ```

  ```
  -vo "output.log"
  ```

  If this option is not given, the output will be shown on terminal.

- **–input_start_dt TEXT**: The start date for the computation of the distributions, must be between the input file date range. (format = "Y-m-d H:M:S")

  The following are two ways to specify that the start date for the computation of the distributions is 2020-1-3 00:00:00 :

  ```
  --input_start_dt "2020-1-3 00:00:00"
  ```

  ```
  -is "2020-1-3 00:00:00"
  ```

  Note: If input start date is not given, it will take the first date of the input xyid file as input start date.

- **–input_end_dt TEXT**: The end date for the computation of the distributions, must be between the input file date range. (format = "Y-m-d H:M:S")

  The following are two ways to specify that the end date for the computation of the distributions is 2020-1-3 00:00:00 :

```
--input_end_dt "2020-1-3 00:00:00"
```

```
-ie "2020-1-3 00:00:00"
```

Note: If input end date is not given, it will take the last date of the input xyid file as input end date.

- **–simulation_start_dt TEXT**:

  The start date of the simulation of scenarios, must be between the input file date range. (format = “Y-m-d H:M:S”)

  The following are two ways to specify that the start date for the simulation is 2020-1-3 00:00:00 :

  ```
  --simulation_start_dt "2020-1-3 00:00:00"
  ```

  ```
  -ss "2020-1-3 00:00:00"
  ```

  Note: If the simulation start date is not given, it will take the first date of the sid file as simulation start date.

- **–simulation_end_dt TEXT**:

  The end date of the simulation of scenarios, must be between the input file date range. (format = “Y-m-d H:M:S”)

  The following are two ways to specify that the end date for the simulation is 2020-1-3 00:00:00 :

  ```
  --simulation_end_dt "2020-1-3 00:00:00"
  ```

  ```
  -se "2020-1-3 00:00:00"
  ```

  Note: If the simulation end date is not given, it will take the last date of the sid file as simulation end date.

- **–target_mape FLOAT**:

  The target MAPE (Mean Absolute Percentage Error) sets the value of the desired MAPE for the simulated data.

  The following are the two ways to specify that the target MAPE should be 41.1:

  ```
  --target_mape 41.1
  ```

  ```
  -t 41.1
  ```

  If this option is not given, the target MAPE is the MAPE of the input data.

- **–a FLOAT**:

  When estimating the conditional beta distribution parameters over a sample, a% of data on the left and a% on the right is used.

  The following are the two ways to specify that the percent of data is 4.3:

  ```
  --a 4.3
  ```

  ```
  -a 4.3
  ```

  If this option is not given, the percent of data on the left and on the right for the estimation is assumed to be 4.

- **–curvature_target FLOAT**:

  Target of the second difference when the user wants to optimize the scenarios curvature.

  The following are two ways to specify that the target of the second difference is 3.1:

  ```
  --curvature_target 3.1
  ```

  ```
  -ct 3.1
  ```

  If this option is not given, the target of the second difference is assumed to be the mean of the second difference of the dataset.

- **–mip_gap FLOAT**:

  Mip gap for curvature optimization

  The following are two ways to specify that the mip gap is 0.1:

  ```
  --mip_gap 0.1
  ```

  ```
  -m 0.1
  ```

  If this option is not given, the mip gap is assumed to be 0.3.

- **–number_simulations INTEGER**:

  The number of scenarios to create.

  The following are the two ways to specify that the number of simulations is 4:

  ```
  --number_simulations 4
  ```

  ```
  -n 4
  ```

  If this option is not given, the number of simulations is assumed to be 1.

- **–time_limit INTEGER**:

  Time limit for curvature optimization.

  The following are two ways to specify that the time limit is 40 seconds:

  `--time_limit 40`

  `-tl 40`

  If this option is not given, the time limits is assumed to be 3600 seconds.

- **–plot_start_date INTEGER**:

  Start date of the plot.

  The following are two ways to specify that the plot start date is the first day:

  `--plot-start_date 0`

  `-ps 0`

  If this option is not given, it is assumed to be 0 and the simulations will be plotted starting from the first date.

- **–seed INTEGER**:

  The seed used for simulation. If none, the seed will be random.

  The following are two ways to specify that the title if the seed is set as "1134":

  `--seed 1134`

  `-s 1134`

  If this option is not given, the seed will be randomly chosen.

- **–verbosity INTEGER**:

  We have 3 options to choose:

  - 2 (logging.INFO), will output info, error, and warning messages.
  - 1 (logging.WARNING), will output error and warning messages.
  - 0 (logging.ERROR), will only output error messages.

  The following are two ways to specify the verbosity level:

  `--verbosity 2`

  `-v 2`

If this option is not given, the verbosity level will set logging.INFO as default.

- **–sid_feature TEXT**:

  If the user wants to simulate actuals from forecasts, then the simulated timeseries will be "actuals". On the other hands, if the user wants to simulate forecasts from actuals, then the simulated timeseries will be "forecasts".

  The following are the two ways to specify that simulated timeseries is "actuals":

  ```
  --sid_feature "actuals"
  ```

  ```
  -f "actuals"
  ```

  If this option is not given, the simulated timeseries is assumed to be "actuals".

- **–base_process TEXT**:

  The base process is a timeseries of random variables with marginal law following a normal law of mean 0 and variance 1. We then apply a transformation to the base process to retrieve the simulated errors. The base process can either be independent and identically distributed ("iid"), or simulated via an ARMA process ("ARMA"). In the last case, the base process will be correlated, hence the errors will have a stronger correlation than with an "iid" base process.

  The following are the two ways to specify that base process is iid:

  ```
  --base_process "iid"
  ```

  ```
  -bp "iid"
  ```

  If this option is not given, the base process is assumed to be "ARMA"

- **–load_pickle BOOLEAN**:

  This will load the pickle file of the estimated parameters for the input dataset and the output feature instead of re-estimating the parameters for the conditional beta distributions.

  This command can be used to improve the speed of the program by skipping the estimation part. However, it can only happen if a previous run was made for the same input dataset and for the same output feature.

  The following are two ways to specify that mape-maker should load the estimated parameters if they exist:

  ```
  --load_pickle
  ```

  ```
  -lp
  ```

  Note: Every run of mape-maker will create a new pickle file or update the existing one for that specific input dataset and output feature. The file is stored in the stored_vectors subdirectory in the mape_maker directory.

If the pickle file does not exist or if this option is not given, then the parameters for the beta distributions are computed.

- **–curvature BOOLEAN**:

  True if the user wants to optimize the scenarios curvature.

  Curvature is the second difference of the time series of output. (If you are not sure whether to use the curvature, you should set it as False)

  The following are two ways to specify that the curvature is True:

  ```
  --curvature
  ```

  ```
  -c
  ```

  If this option is not given, the curvature is assumed to be False

- **–show_curv_model BOOLEAN**:

  True if the user wants to show the model for curvature.

  The following are two ways to specify to show the model:

  ```
  --show_curv_model
  ```

  ```
  -sh
  ```

  If this option is not given, the option is assumed to be False

- **–plot BOOLEAN**:

  True if the user wants to plot the results.

  The following are two ways to specify to not plot the result:

  ```
  --plot
  ```

  ```
  -p
  ```

  If this option is not given, the option is assumed to be True

- **–solver TEXT**:

  The name of the software that is used to perform the curvature optimization process.

  The following are two ways to specify that the solver is “cplex”:

  ```
  --solver "cplex"
  ```

```
-sv "cplex"
```

If this option is not given, the solver is assumed to be “gurobi”.

- **–title TEXT**:

  The title of the simulation plot.

  The following are two ways to specify the title of the simulation plot as “my plot”:

  ```
  --title "my plot"
  ```

  ```
  -tt "my plot"
  ```

  If this option is not given, the title of the simulation plot is assumed to be None. Therefore, no additional title will be added to the plot.

- **–x_legend TEXT**:

  The x legend of the simulation plot.

  The following are two ways to specify the x legend of the simulation plot as “x legend”:

  ```
  --x_legend "x legend"
  ```

  ```
  -xl "x legend"
  ```

  If this option is not given, the x legend of the simulation plot is assumed to be None. Therefore, no additional legend will be added to the plot.

- **–scale_by_capacity FLOAT**:

  Calculate MAPE relative to capacity instead of observations, i.e.

  $$mape = \frac{100}{n}\sum_{i=1}^{n}\frac{|f_i - a_i|}{cap}$$

  The following are the two ways to specify that the capacity is 2000:

  ```
  --scale_by_capacity 2000
  ```

  ```
  -sb 2000
  ```

  If this option is not given, scale by observation.

  If this option is given to be 0, capacity is set to be the maximum of the observation.

- **–target_scaled_capacity FLOAT**:

Optionally enter target capacity to scale all simulated data by target_capacity/capacity

The following are the two ways to specify that the target capacity is 1000:

```
--target_scaled_capacity 1000
```

```
-ts 1000
```

If this option is not given, simulated data is not scaled.

4.2 By Default-options

- **input_sid_file** : None, will take the input dataset as sid
- **output_dir** : None, no output_file will be created while a plot will be outputted
- **verbosity_output** : None, no verbosity_output will be created while a plot will be outputted
- **input_start_dt** : None, will use the whole dataset for the computation of the distributions
- **input_end_dt** : None, will use the whole dataset for the computation of the distributions
- **simulation_start_dt** : None, will simulate over the whole dataset
- **simulation_end_dt** : None, will simulate over the whole dataset
- **target_mape** : the mape of the current dataset
- **a** : 4
- **curvature_target** : mean of the second difference of the dataset
- **mip_gap** : 0.3
- **number_simulations** : 1
- **time_limit** : 3600 seconds
- **plot_start_date** : 0
- **seed** : 1234
- **verbosity** : 2
- **sid_feature** : “actuals”
- **base_process** : “ARMA”
- **load_pickle** : False
- **curvature** : False
- **show_curv_model** : False
- **plot** : True
- **solver** : gurobi
- **title** : None, no additional title will be added to the plot
- **x_legend** : None, will use the feature of curves (actuals or forecasts)
- **scale_by_capacity** : None, will not scale by capacity
- **target_scaled_capacity**: None, will not scale simulated data

CHAPTER 5

CAISO wind data file examples

Sample Command 1:

The following command will take the data from *wind_total_forecast_actual_070113_063015.csv*, and launch the simulations with n=3 and s=1234 from forecasts to actuals using an iid Base Process. It will compute the distribution from 2014-7-1 00:00:00 to 2014-8-1 00:00:00 and simulate from 2014-7-2 00:00:00 to 2014-7-31 00:00:00. Finally, it will return a plot of simulations, and create an output dir called “wind_actuals_iid” in your current working directory.

```
python -m mape_maker -xf "mape_maker/samples/wind_total_forecast_actual_070113_063015.
↪csv" -f "actuals" -n 3 -bp "iid" -o "wind_actuals_iid" -is "2014-7-1 00:00:00" -ie
↪"2014-8-1 00:00:00" -ss "2014-7-2 00:00:00" -se "2014-7-31 00:00:00" -s 1234
```

- **-xf “mape_maker/samples/wind_total_forecast_actual_070113_063015.csv”**:

  The csv file containing forecasts and actuals for specified datetimes.

- **-f “actuals”**:

  Set up the the target of the simulation as “actuals”. So the MapeMaker will simulate the “actuals” data according to the “forecasts” data in the input file.

- **-n 3**:

  The number of simulations that we want to create is “3”. This will create three simulation columns in the output file.

- **-bp “iid”**:

  Use “iid” as the base process. The default base process is set as “ARMA”.

- **-is “2014-7-1 00:00:00”**:

  The start date for the computation of the distributions is “2014-7-1 00:00:00”

- **-ie “2014-8-1 00:00:00”**:

  The end date for the computation of the distributions is “2014-8-1 00:00:00”

- **-ss “2014-7-2 00:00:00”**:

The start date of the simulation is “2014-7-2 00:00:00”

- **-se “2014-7-31 00:00:00”**:

  The end date of the simulation is “2014-7-31 00:00:00”

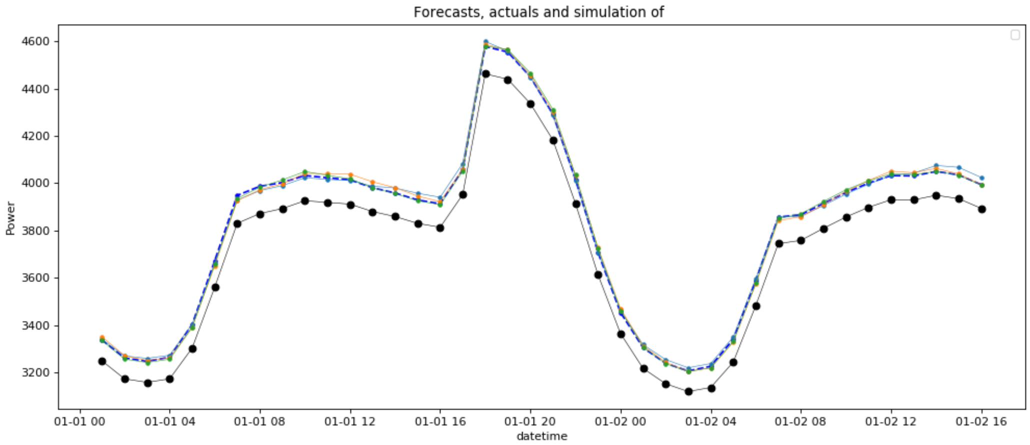
- **-s 1234**:

  Set the seed as “1234”, so it won’t randomly choose a number as the seed.

- **-o “wind_actuals_iid”**:

  Create an output directory called “wind_actuals_iid”, in which will store the simulation output file.

After running the command line, you should see a similar plot like this:



Sample Command 2:

The following command will take the data from *wind_total_forecast_actual_070113_063015.csv* , and launch the simulations with n=3 and seed=1234 from forecasts to actuals using an ARMA Base Process. It will simulate all the dates in the input files. Finally, it will return a plot of simulations, and create an output dir called “wind_actuals_ARMA”.

```
python -m mape_maker -xf "mape_maker/samples/wind_total_forecast_actual_070113_063015.
↪csv" -f "actuals" -n 3 -bp "ARMA" -o "wind_actuals_ARMA" -s 1234
```

- **-xf “mape_maker/samples/wind_total_forecast_actual_070113_063015.csv”**:

  The csv file containing forecasts and actuals for specified datetimes.

- **-f “actuals”**:

  Set up the the target of the simulation as “actuals”. So the MapeMaker will simulate the “actuals” data according to the “forecasts” data in the input file.

- **-n 3**:

The number of simulations that we want to create is "3". This will create three simulation columns in the output file.

- **-bp "ARMA"**:

  Use "ARMA" as the base process.

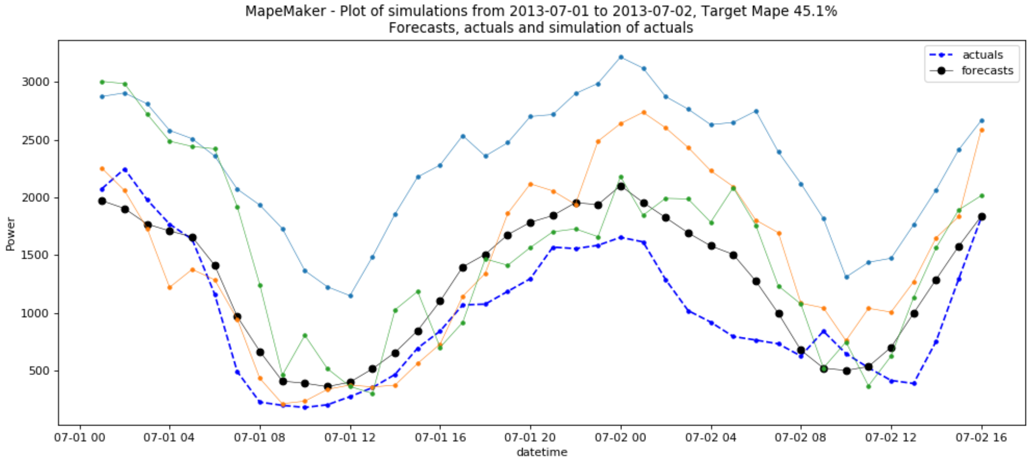
- **-o "wind_actuals_ARMA"**:

  Create an output directory called "wind_actuals_ARMA", in which will store the simulation output file.

- **-s 1234**:

  Set the seed as "1234", so it won't randomly choose a number as the seed.

After running the command line, you should see a similar plot like this:



CHAPTER 6

Demand data file examples

The following command will take the data based on rts gmlc's Load time series and launch the simulations with n = 3 and seed = 1234 from forecasts to actuals using an ARMA Base Process. It will simulate all the dates in the input files. Finally, it will return a plot of simulations, and create an output dir called "load_actuals_ARMA".

```
python -m mape_maker -xf "mape_maker/samples/based_rts_gmlc/Load_rts_gmlc_based/
↪processed_file.csv" -f "actuals" -n 3 -bp "ARMA" -o "load_actuals_ARMA" -s 1234
```

- **-xf "mape_maker/samples/based_rts_gmlc/Load_rts_gmlc_based/processed_file.csv"**:

  The csv file containing forecasts and actuals for specified datetimes.

- **-f "actuals"**:

  Set up the the target of the simulation as "actuals". So the MapeMaker will simulate the "actuals" data according to the "forecasts" data in the input file.

- **-n 3**:

  The number of simulations that we want to create is "3". This will create three simulation columns in the output file.

- **-bp "ARMA"**:

  Use "ARMA" as the base process. The default base process is set as "ARMA".

- **-o "load_actuals_ARMA"**:

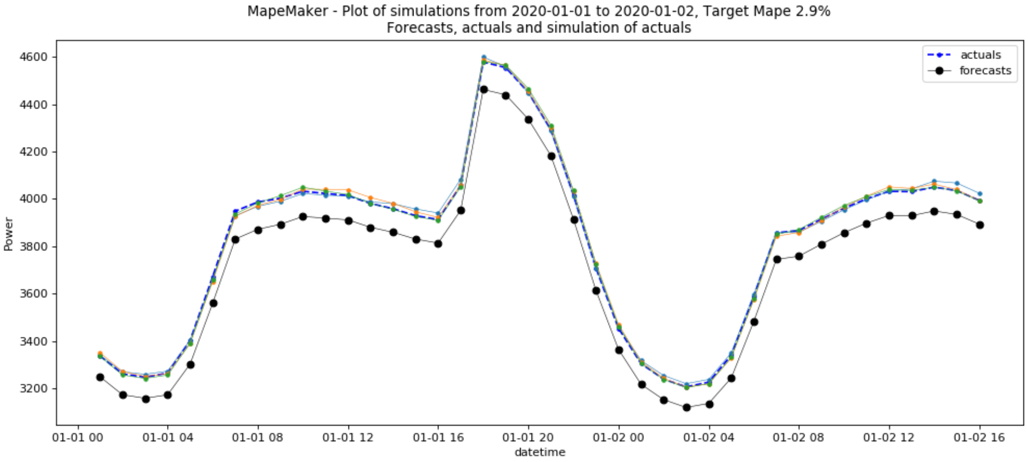
  Create an output directory called "load_actuals_ARMA", in which will store the simulation output file.

- **-s 1234**:

  Set the seed as "1234", so it won't randomly choose a number as the seed.

After running the command line, you should see a similar plot like this:

Since rts_gmlc Load data has very little relative error and hence very little mape, the scenario lines tend to overlap in the plot.


MapeMaker - Plot of simulations from 2020-01-01 to 2020-01-02, Target Mape 2.9%
Forecasts, actuals and simulation of actuals
actuals
forecasts
Power
4600
4400
4200
4000
3800
3600
3400
3200
datetime
01-01 00
01-01 04
01-01 08
01-01 12
01-01 16
01-01 20
01-02 00
01-02 04
01-02 08
01-02 12
01-02 16

CHAPTER 7

Operational Example

7.1 CAISO Wind Operations Examples

The following command will take the data from *CAISO_wind_operational_data.csv.csv*, which is a modified version of CAISO's *wind_total_forecast_actual_070113_063015.csv*. We will find the scenarios from forecasts to actuals for the date(s) that have missing actuals.

It will simulate all the dates in the input files, and can use the date(s) with missing actuals for simulation dates. In order to fit the process well, we need to simulate for a few additional hours before the day we are interested in as the ARMA process can be a little imprecise for the first few hours. For convenience/efficiency, we will add the preceding day to the desired simulation start date. Here, for example, we are interested in 2015-6-30, but we will use 2015-6-29 as the simulation start date. Finally, it will return a plot of simulations, and create an output dir called "Wind_Operation1".

```
python -m mape_maker -xf "mape_maker/samples/CAISO_wind_operational_data.csv" -s 1234␣
↪-n 5 -bp "ARMA" -o "Wind_Operation1" -is "2013-7-1 00:00:00" -ie "2015-6-30 23:00:00
↪" -ss "2015-6-29 23:00:00" -se "2015-6-30 23:00:00"
```

- **-xf "mape_maker/samples/CAISO_wind_operational_data.csv"**:

  The csv file containing forecasts and actuals for specified datetimes.

- **-n 5**:

  The number of simulations that we want to create is "3". This will create three simulation columns in the output file.

- **-bp "ARMA"**:

  Use "ARMA" as the base process. The default base process is set as "ARMA".

- **-is "2013-7-1 00:00:00"**:

  The start date for the computation of the distributions is "2013-7-1 00:00:00"

- **-ie "2015-6-30 23:00:00"**:

  The end date for the computation of the distributions is "2015-6-30 23:00:00"

- **-ss "2015-6-29 23:00:00"**:

  The start date of the simulation is "2015-6-29 23:00:00"

- **-se "2015-6-30 23:00:00"**:

  The end date of the simulation is ""2015-6-30 23:00:00"

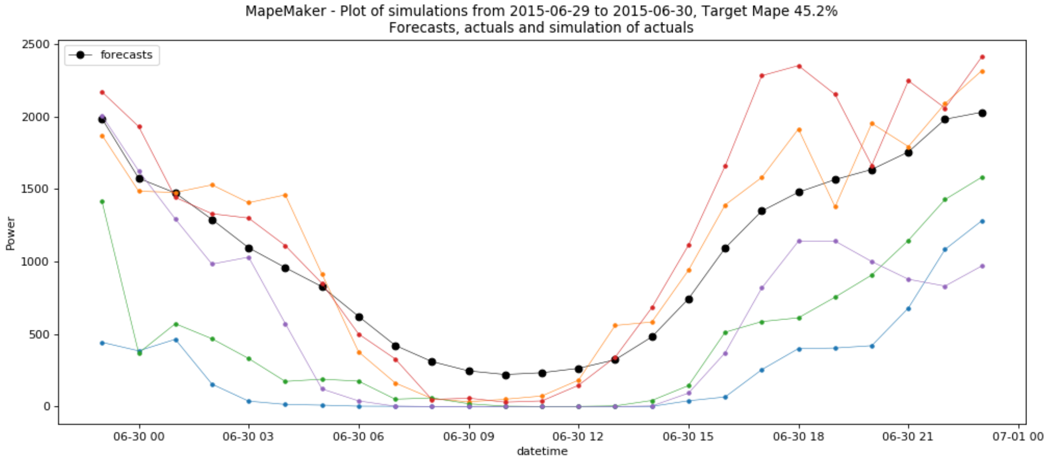
- **-o "Wind_Operation1"**:

  Create an output directory called "Wind_Operation1", in which will store the simulation output file.

- **-s 1234**:

  Set the seed as "1234", so it won't randomly choose a number as the seed.

After running the command line, you should see a similar plot like this:



CHAPTER 8

Second File Example

8.1 CAISO Wind and BPA data Examples

The user need to set the second file option when the file that he/she want to simulate is missing forecast/actual data. The following command will take the input file *2012-2013_BPA_forecasts_actuals.csv* and the second file *wind_total_forecast_actual_070113_063015.csv*. The second file will learn the error distribution from the input file, and use it to create the simulation data. In order to fit the process well, we used a wild range of data for the input file. In this example, we set the date range for the input file from "2012-6-3 00:00:00" to "2013-8-3 00:00:00 and generate the scenarios from "2015-6-29 00:00:00" to "2015-6-30 00:00:00" for the second file.

```
python -m mape_maker -xf "mape_maker/samples/2012-2013_BPA_forecasts_actuals.csv" -sf
↪"mape_maker/samples/wind_total_forecast_actual_070113_063015.csv" -s 1234 -n 5 -o
↪"BPA_Wind_1" -is "2012-6-3 00:00:00" -ie "2013-8-3 00:00:00" -ss "2015-6-23 00:00:00
↪" -se "2015-6-30 00:00:00"
```

- **-xf "mape_maker/samples/2012-2013_BPA_forecasts_actuals.csv"**:

  The input file containing forecasts and actuals for specified datetimes.

- **-sf "mape_maker/samples/wind_total_forecast_actual_070113_063015.csv"**

  The second file containing forecasts and actuals for specified datetimes.

- **-n 5**:

  The number of simulations that we want to create is "3". This will create three simulation columns in the output file.

- **-bp "ARMA"**:

  Use "ARMA" as the base process. The default base process is set as "ARMA".

- **-is "2012-6-3 00:00:00"**:

  The start date for the computation of the distributions is "2012-6-3 00:00:00"

- **-ie "2013-8-3 00:00:00"**:

The end date for the computation of the distributions is “2013-8-3 00:00:00”

- **-ss “2015-6-23 00:00:00”**:

  The start date of the simulation is “2015-6-23 00:00:00”

- **-se “2015-6-30 00:00:00”**:

  The end date of the simulation is “2015-6-30 00:00:00”

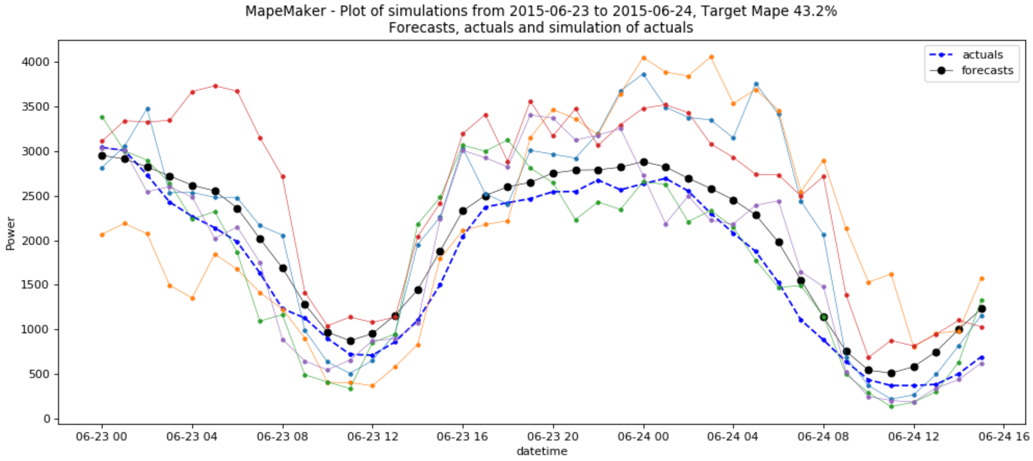
- **-o “BPA_Wind_1”**:

  Create an output directory called “BPA_Wind_1”, in which will store the simulation output file.

- **-s 1234**:

  Set the seed as “1234”, so it won’t randomly choose a number as the seed.

After running the command line, you should see a similar plot like this:



CHAPTER 9

Using rts_gmlc data file as input for MapeMaker

MapeMaker can be used with rts_gmlc data files found on their github website. The data files need to be processed into the required 3 columns format - datetime, actuals, and forecasts. We use “process_RTS_GMLC_data_s.py” file for this. Here is a step-by-step explanation:

1. Git clone the rts_gmlc data files to your working directory. (The link might be updated, please modify it accordingly)

```
git clone https://github.com/GridMod/RTS-GMLC.git
```

2. Now use the process_RTS_GMLC_data_s.py file to process the data in the required format for MapeMaker. First, cd to the directory with the program:

```
cd mape_maker/samples/rts_gmlc/
```

Then, you can run the python script as follows:

```
python process_RTS_GMLC_data_s.py timeseries_path source_path write_path
```

where

- **timeseries_path**: location of the timeseries_data_files directory (e.g. “RTS-GMLC/RTS_Data/timeseries_data_files/”)
- **source_path**: location of the SourceData directory (e.g. “RTS-GMLC/RTS_Data/SourceData/”)
- **write_path**: location of an existing directory you want to store the processed files in (e.g. “my_rts_gmlc”)

After running the python script, the write_path will contain all csv and txt files for all the timeseries data files processed by buses, zones, and aggregated together over all the zones. The csv files can be used as input to MapeMaker to get the desired scenarios.

3. Some of the rts_gmlc data files entries have a lot more than acceptable errors. If you want to modify the data based on a threshold, such that if the absolute relative error for a particular datetime is higher than the threshold, change the forecast so that the absolute relative error for that datetime equals threshold. In order to adapt you data file go to:

```
cd mape_maker/samples/based_rts_gmlc/
```

Then, you can run the python script as follows:

```
python process_based_on_RTS_GMLC.py source_path write_path threshold
```

where

- **source_path**: location of the timeseries_data_files directory (e.g. “RTS-GMLC/RTS_Data/timeseries_data_files/”)
- **write_path**: location of an existing directory you want to store the processed files in (e.g. “adapted_rts_gmlc”)
- **threshold**: the value used to modify the dataset if needed

After running the python script, the write_path will contain the processed csv file that can be used as input to Mape-Maker to get the desired scenarios.

For convenience, we will use the adapted rts_gmlc data files. Here are some examples.

9.1 Example 1 - WIND_forecasts_actuals.csv

The following command will take the data from rts_gmlc based Wind data file, and launch the simulations with n = 5 and seed = 1234 from forecasts to actuals using an ARMA base process. It will compute the distribution from 2020-2-1 00:00:00 to 2020-5-1 00:00:00 and simulate from 2020-2-2 00:00:00 to 2020-3-2 00:00:00. Finally, it will return a plot of simulations, and create an output dir called “wind_forecasts_actuals”.

```
python -m mape_maker -xf "mape_maker/samples/based_rts_gmlc/Wind_rts_gmlc_based/
↪processed_file.csv" -f "actuals" -s 1234 -n 5 -bp "ARMA" -o "wind_forecasts_actuals
↪" -is "2020-2-1 00:00:00" -ie "2020-5-1 00:00:00" -ss "2020-2-2 00:00:00" -se "2020-
↪3-2 00:00:00"
```

- **-xf “mape_maker/samples/based_rts_gmlc/Wind_rts_gmlc_based/processed_filecsv”**:

  The csv file containing forecasts and actuals for specified datetime.

- **-f “actuals”**:

  Set up the the target of the simulation as “actuals”. So the MapeMaker will simulate the “actuals” data according to the “forecasts” data in the input file.

- **-n 5**:

  The number of simulations that we want to create is “5”. This will create three simulation columns in the output file.

- **-bp “ARMA”**:

  Use “ARMA” as the base process.

- **-o “wind_forecasts_actuals”**:

  Create an output directory called “wind_forecasts_actuals” in the temporary subdirectory, in which will store the simulation output file. The program prints the temporary subdirectory in the temporary directory. The user can retrieve the output dir using that location.

- **-s 1234**:

  Set the seed as “1234”, so it won’t randomly choose a number as the seed.

- **-is “2020-2-1 00:00:00”**:

The start date for the computation of the distributions is “2020-2-1 00:00:00”

- **-ie “2020-5-1 00:00:00”**:

  The end date for the computation of the distributions is “2020-5-1 00:00:00”

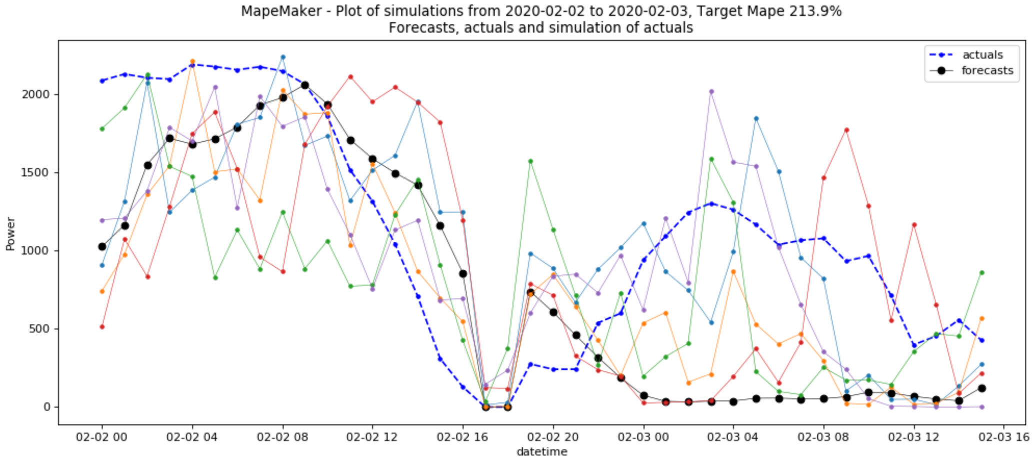
- **-ss “2020-2-2 00:00:00”**:

  The start date of the simulation is “2020-2-2 00:00:00”

- **-se “2020-3-2 00:00:00”**:

  The end date of the simulation is “2020-3-2 00:00:00”

After running the command line, you should see a similar plot like this:



9.2 Example 2 - Bus_220_Load_zone2_forecasts_actuals.csv

The following command will take the data from *Bus_220_Load_zone2_forecasts_actuals.csv*, and launch the simulations with n = 5 and seed = 1234 from forecasts to actuals using an ARMA base process. It will simulate all the dates in the input files. Finally, it will return a plot of simulations, and create an output dir called “Bus_220_load”.

```
python -m mape_maker -xf "mape_maker/samples/rts_gmlc/Bus_220_Load_zone2_forecasts_
↪actuals.csv" -f "actuals" -n 5 -bp "ARMA" -is "2020-1-10 1:0:0" -ie "2020-7-20 0:0:0
↪" -ss "2020-6-1 0:0:0" -se "2020-6-30 23:0:0" -o "Bus_220_load" -s 1234
```

- **-xf “mape_maker/samples/rts_gmlc/Bus_220_Load_zone2_forecasts_actuals.csv”**:

  The csv file containing forecasts and actuals for specified datetimes.

- **-f “actuals”**:

  Set up the the target of the simulation as “actuals”. So the MapeMaker will simulate the “actuals” data according to the “forecasts” data in the input file.

- **-n 5**:

  The number of simulations that we want to create is “5”. This will create three simulation columns in the output file.

- **-bp “ARMA”**:

  Use “ARMA” as the base process. The default base process is set as “ARMA”.

- **-is “2020-1-10 1:0:0”**:

  The start date of the input data for processing is “2020-1-10 1:0:0”
- **-ie “2020-7-20 0:0:0”**:

  The end date of the input data for processing is “2020-7-20 0:0:0”
- **-ss “2020-6-1 0:0:0”**:

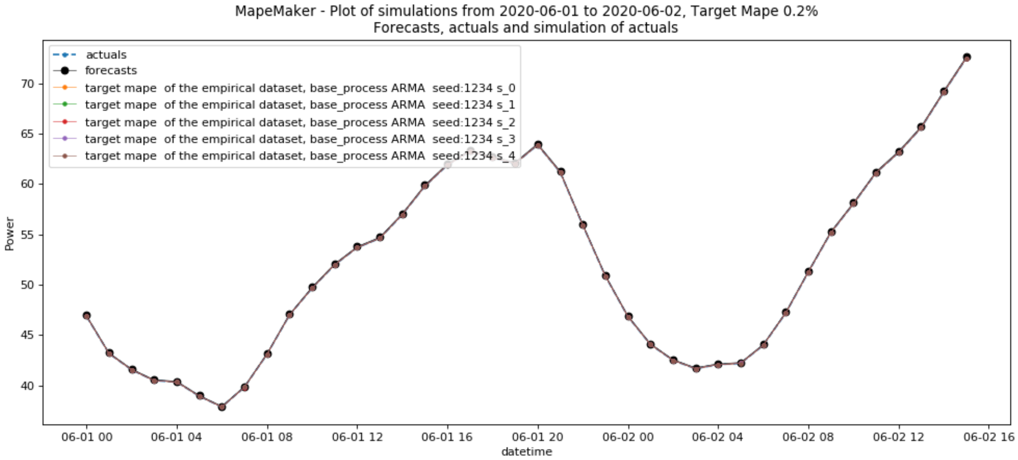
  The start date of the scenario simulation is “2020-6-1 0:0:0”
- **-ed “2020-6-30 23:0:0”**:

  The end date of the scenario simulation is “2020-6-30 23:0:0”
- **-o “Bus_220_load”**:

  Create an output directory called “Bus_220_load”, in which will store the simulation output file.
- **-s 1234**:

  Set the seed as “1234”, so it won’t randomly choose a number as the seed.

After running the command line, you should see a similar plot like this:



Since rts_gmlc Load data has very little relative error and hence very little mape, the scenario lines tend to overlap in the plot.

CHAPTER 10

Unit Tests

10.1 Unittest Examples for Wind and Load

CAISO Wind Unittest Examples

---

fast_CAISO_wind_tests.py

- This test will run very fast because it only includes one sample command line.

quick_CAISO_wind_tests.py

- This test is used to check whether some sample command lines will run successfully at once.

slow_CAISO_wind_tests.py

- The difference between CAISO_wind_quick_test.py and CAISO_wind_slow_test.py is that the second one will compare the first simulation column and the second simulation column, then show the differences.

Load Unittest Examples

- This test will run very fast because it only includes one sample command line.

load_quick_test.py

- This test is used to check whether some sample command lines will run successfully at once.

load_slow_test.py

- The difference between load_quick_test.py and load_slow_test.py is that the second one will compare the first simulation column and the second simulation column, then show the differences.

cap_test.py

- This test is used to check whether the scale_by_capacity option works properly.

cap_scale_test.py

- This test is used to check whether the target_scale_capacity option works properly by comparing the data scaled by the input option and the manually scaled data.

10.2 Unittest Examples for RTS WIND and BUS

RTS Wind Unittest Examples

---

rts_wind_test.py

- This test is used to see whether the sample command line for RTS Wind example will run successfully.

Bus_220_Load_test.py

- This test is used to see whether the sample command line for Load example will run successfully.

10.3 Unittest Examples for xxx_makers

BPA_maker_test.py

- This test is used to see whether the BPA_maker will run successfully.

fake_BPA_maker_test.py

- This test is used to see whether the fake_BPA_maker will run successfully.

CAISO_maker_test.py

- This test is used to see whether the CAISO_maker will run successfully.

CHAPTER 11

Infeasible_example

For any requested MAPE, the distributions of errors computed should be close enough to the input error distributions while meeting the target MAPE. During the process, we compute the mean absolute error of a conditional beta distribution with fixed alpha and beta to satisfy the target MAPE so that it keeps the same shape parameters(denoted by alpha and beta) as the original distribution. If it's hard to meet the target MAPE, then it will throw the error, saying that it's infeasible to meet the target MAPE.

11.1 Example 1

```
python -m mape_maker -xf "mape_maker/samples/wind_total_forecast_actual_070113_063015.
↪csv" -f "actuals" -n 5 -bp "ARMA" -ss "2014-6-17 01:00:00" -se "2014-6-30 00:00:00"␣
↪-s 1234
```

This command line will fail with an error, showing the following output:

```
RuntimeError:  < 1, there is a prevalence of high power input in the SID
The requested r_tilde is too high
     => Either change your requested mape to be less than 21.473508488457284
     => Or change your SID so the e_score increases
```

The plausibility score should be close to 1, meaning that the error distribution for the set is close to the empirical distribution of errors. In this example, we can see that the maximum mare is less than the target mare. In order to make the program run successfully, the user can either change the target mape or adjust the input dataset.

11.2 Example 2

```
python -m mape_maker -xf "mape_maker/samples/wind_total_forecast_actual_070113_063015.
↪csv" -f "actuals" -n 5 -bp "ARMA" -o "wind_actuals_ARMA_1" -is "2014-6-1 00:00:00" -
↪ie "2014-6-30 00:00:00" -ss "2014-6-15 01:00:00" -se "2014-6-29 00:00:00" -s 1234
```

This command line will fail with an error, showing the following output during the running process: *** WARNING!! ******

> beta rvs failed at i=684,x=3314.9; a=-0.05177452493780477, b=-0.3897512549813409, l_=-292.78035, s_=359.6603500000001 Using last good beta parameters.

This error occurs when the program is trying to estimate the maximum target mean absolute error function(called as m_max). It fails because the shape parameters(alpha and beta) are negative values. .. note:: If there have not been any good beta parameters, the program will terminate, but otherwise, it will continue.

CHAPTER 12

solar_mape_maker

The `solar_mape_maker.py` program enables creation of scenario files using solar data. It calculates the difference between the input data and upper bound of solar energy generation at the same time as the input data. The upper bound uses the maximum of clear-sky POA from the location parameters users put in, and the algorithm is described in *Constructing probabilistic scenarios for wide-area solar power generation* by David L. Woodruff, et al. (<https://doi.org/10.1016/j.solener.2017.11.067>, see citing section for more information).

This difference is used to feed in `Mape_Maker` to generate scenarios. Then the output of `Mape_Maker` is added to the upper bound to create the final solar scenarios.

12.1 Required Arguments

- **–input_solar_file TEXT**:

  The path to input solar dataset that contains date time, forecasts and actuals (the same format as the input of `Mape_Maker`).

  The following specify "solar_data.csv" as the input file:

  `--input_solar_file "solar_data.csv"`

  `-isf "solar_data.csv"`

- **—location_coor int**:

  The coordinates of the location where the input solar data is collected. It can be either a pair of coordinates for data collected from an individual site, or several pairs of coordinates that are the extreme points (northernmost/southernmost/easternmost/westernmost) of the group of sites from which the data is collected. Use space to separate numbers and enter in the sequence of latitude_1 longitude_1 latitude_2 longitude_2

The following specify the location as (37N 103W):

```
--location_coor "37 -103"
```

```
-lc "37 -103"
```

The following specify the rage of generation site location is within (37N 103W), (31N 94W) and (32N 107W)

```
--location_coor "37 -103 31 -94 26 -98 32 -107"
```

```
-lc "37 -103 31 -94 26 -98 32 -107"
```

12.2 Options

- **–input_sid_file TEXT**:

  The path to a simulation input dataset (sid) with one or two timeseries (e.g. actuals), from which scenarios for the other timeseries are generated (e.g. forecasts)

  The following loads "sid.csv" located under the current directory :

  ```
  --input_sid_file "sid.csv"
  ```

  ```
  -sf "sid.csv"
  ```

  If this option is not given, the sid will be taken as a subset of the input dataset, specified by a simulation_start_dt and simulation_end_dt.

- **–solar_output TEXT**:

  Path to destination dir where the scenario are saved as csv file(s).

  The following are the two ways to specify that the output directory is called "output":

  ```
  --solar_output "output"
  ```

  ```
  -so "output"
  ```

  If this option is not given, the output directory is assumed to be None. No output directory will be created.

  Note: If the output directory is not given, then the only output will be a png image of the plot showing the scenarios and saved under the current directory.

  Warning: If the output directory already exists, the program will terminate and issue messages. It won't overwrite an existing directory.

- **–verbosity_output TEXT**:

  The name of the verbosity output file

  The following are two ways to specify the verbosity level:

```
--verbosity_output "output.log"
```

```
-vo "output.log"
```

If this option is not given, the output will be shown on terminal.

- **–input_end_dt TEXT**: The end date for the computation of the distributions, must be between the input file date range. (format = "Y-m-d H:M:S")

  The following are two ways to specify that the end date for the computation of the distributions is 2020-1-3 00:00:00 :

  ```
  --input_end_dt "2020-1-3 00:00:00"
  ```

  ```
  -ie "2020-1-3 00:00:00"
  ```

  Note: If input end date is not given, it will take the last date of the input xyid file as input end date.

- **–simulation_start_dt TEXT**:

  The start date of the simulation of scenarios, must be between the input file date range. (format = "Y-m-d H:M:S")

  The following are two ways to specify that the start date for the simulation is 2020-1-3 00:00:00 :

  ```
  --simulation_start_dt "2020-1-3 00:00:00"
  ```

  ```
  -ss "2020-1-3 00:00:00"
  ```

  Note: If the simulation start date is not given, it will take the first date of the sid file as simulation start date.

- **–simulation_end_dt TEXT**:

  The end date of the simulation of scenarios, must be between the input file date range. (format = "Y-m-d H:M:S")

  The following are two ways to specify that the end date for the simulation is 2020-1-3 00:00:00 :

  ```
  --simulation_end_dt "2020-1-3 00:00:00"
  ```

  ```
  -se "2020-1-3 00:00:00"
  ```

  Note: If the simulation end date is not given, it will take the last date of the sid file as simulation end date.

- **–target_mape FLOAT**:

  The target MAPE (Mean Absolute Percentage Error) sets the value of the desired MAPE for the simulated data.

  The following are the two ways to specify that the target MAPE should be 41.1:

  ```
  --target_mape 41.1
  ```

  ```
  -t 41.1
  ```

  If this option is not given, the target MAPE is the MAPE of the input data.

- **–a FLOAT**:

  When estimating the conditional beta distribution parameters over a sample, a% of data on the left and a% on the right is used.

  The following are the two ways to specify that the percent of data is 4.3:

  ```
  --a 4.3
  ```

  ```
  -a 4.3
  ```

  If this option is not given, the percent of data on the left and on the right for the estimation is assumed to be 4.

- **–curvature_target FLOAT**:

  Target of the second difference when the user wants to optimize the scenarios curvature.

  The following are two ways to specify that the target of the second difference is 3.1:

  ```
  --curvature_target 3.1
  ```

  ```
  -ct 3.1
  ```

  If this option is not given, the target of the second difference is assumed to be the mean of the second difference of the dataset.

- **–mip_gap FLOAT**:

  Mip gap for curvature optimization

  The following are two ways to specify that the mip gap is 0.1:

  ```
  --mip_gap 0.1
  ```

  ```
  -m 0.1
  ```

  If this option is not given, the mip gap is assumed to be 0.3.

- **–number_simulations INTEGER**:

  The number of scenarios to create.

  The following are the two ways to specify that the number of simulations is 4:

  ```
  --number_simulations 4
  ```

  ```
  -n 4
  ```

  If this option is not given, the number of simulations is assumed to be 1.

- **–time_limit INTEGER**:

  Time limit for curvature optimization.

  The following are two ways to specify that the time limit is 40 seconds:

  ```
  --time_limit 40
  ```

  ```
  -tl 40
  ```

  If this option is not given, the time limits is assumed to be 3600 seconds.

- **–seed INTEGER**:

  The seed used for simulation. If none, the seed will be random.

  The following are two ways to specify that the title if the seed is set as “1134”:

  ```
  --seed 1134
  ```

  ```
  -s 1134
  ```

  If this option is not given, the seed will be randomly chosen.

- **–verbosity INTEGER**:

  We have 3 options to choose:

  - 2 (logging.INFO), will output info, error, and warning messages.
  - 1 (logging.WARNING), will output error and warning messages.
  - 0 (logging.ERROR), will only output error messages.

  The following are two ways to specify the verbosity level:

  ```
  --verbosity 2
  ```

  ```
  -v 2
  ```

If this option is not given, the verbosity level will set logging.INFO as default.

- **–sid_feature TEXT**:

  If the user wants to simulate actuals from forecasts, then the simulated timeseries will be "actuals". On the other hands, if the user wants to simulate forecasts from actuals, then the simulated timeseries will be "forecasts".

  The following are the two ways to specify that simulated timeseries is "actuals":

  ```
  --sid_feature "actuals"
  ```

  ```
  -f "actuals"
  ```

  If this option is not given, the simulated timeseries is assumed to be "actuals".

- **–base_process TEXT**:

  The base process is a timeseries of random variables with marginal law following a normal law of mean 0 and variance 1. We then apply a transformation to the base process to retrieve the simulated errors. The base process can either be independent and identically distributed ("iid"), or simulated via an ARMA process ("ARMA"). In the last case, the base process will be correlated, hence the errors will have a stronger correlation than with an "iid" base process.

  The following are the two ways to specify that base process is iid:

  ```
  --base_process "iid"
  ```

  ```
  -bp "iid"
  ```

  If this option is not given, the base process is assumed to be "ARMA"

- **–load_pickle BOOLEAN**:

  This will load the pickle file of the estimated parameters for the input dataset and the output feature instead of re-estimating the parameters for the conditional beta distributions.

  This command can be used to improve the speed of the program by skipping the estimation part. However, it can only happen if a previous run was made for the same input dataset and for the same output feature.

  The following are two ways to specify that mape-maker should load the estimated parameters if they exist:

  ```
  --load_pickle
  ```

  ```
  -lp
  ```

  Note: Every run of mape-maker will create a new pickle file or update the existing one for that specific input dataset and output feature. The file is stored in the stored_vectors subdirectory in the mape_maker directory.

If the pickle file does not exist or if this option is not given, then the parameters for the beta distributions are computed.

- **–curvature BOOLEAN**:

  True if the user wants to optimize the scenarios curvature.

  Curvature is the second difference of the time series of output. (If you are not sure whether to use the curvature, you should set it as False)

  The following are two ways to specify that the curvature is True:

  ```
  --curvature
  ```

  ```
  -c
  ```

  If this option is not given, the curvature is assumed to be False

- **–show_curv_model BOOLEAN**:

  True if the user wants to show the model for curvature.

  The following are two ways to specify to show the model:

  ```
  --show_curv_model
  ```

  ```
  -sh
  ```

  If this option is not given, the option is assumed to be False

- **–solar_plot BOOLEAN**:

  True if the user wants to plot the results.

  The following are two ways to specify to plot the result:

  ```
  --solar_plot
  ```

  ```
  -sp
  ```

  If this option is not given, the option is assumed to be False

- **–solver TEXT**:

  The name of the software that is used to perform the curvature optimization process.

  The following are two ways to specify that the solver is “cplex”:

  ```
  --solver "cplex"
  ```

```
-sv "cplex"
```

If this option is not given, the solver is assumed to be "gurobi".

- **–solar_target_scaled_capacity FLOAT**:

  Optionally enter target capacity to scale all simulated data by target_capacity/capacity

  The following are the two ways to specify that the target capacity is 100:

  ```
  --solar_target_scaled_capacity 100
  ```

  ```
  -sts 100
  ```

  If this option is not given, simulated data is not scaled.

12.3 Example

```
python -m mape_maker.solar.solar_mape_maker -isf "mape_maker/solar/NREL_solar_data.csv
↪" -so "solar_test_output" -n 3 -is "2018-07-01 00:00:00" -ie "2018-12-01 00:00:00" -
↪ss "2018-07-01 00:00:00" -se "2018-07-07 00:00:00" -n 2 -bp "iid" -lc "37 -103 31 -
↪94 26 -98 32 -107" -so "solar_test_output" -sts 100 -sp
```

- **-isf "mape_maker/solar/NREL_solar_data.csv"**:

  The csv file of NREL solar data at the system level for Texas 7k, containing forecasts and actuals from 2018-01-01 to 2018-12-31.

- **-so "solar_test_output"**:

  Create an output directory called "solar_test_output", in which will store the simulation output file.

- **-n 2**:

  The number of simulations that we want to create is "2". This will create two simulation columns in the output file.

- **-is "2018-07-01 00:00:00"**:

  The start time of the simulation is "2018-07-01 00:00:00".

- **-ie "2018-12-01 00:00:00"**:

  The end time of the simulation is "2018-12-01 00:00:00".

- **-ss "2018-07-01 00:00:00"**:

  The start time of the simulation is "2018-07-01 00:00:00".

- **-se "2018-07-07 00:00:00"**:

  The end time of the simulation is "2013-07-07 00:00:00".

- **-bp "iid"**:

  Use "iid" as the base process. The default base process is set as "ARMA".

- **-lc "37 -103 31 -94 26 -98 32 -107"**:

  Specify the rage of generation site location is within (37N 103W), (31N 94W) and (32N 107W).
- **-sts 100**:

  Specify the target capacity is 100, and scale all scenario data by target_capacity/capacity, where capacity is the max of observation.
- **-sp**:

  Plot the output.

12.3.1 Default option values

- **input_sid_file** : None, will take the input dataset as sid
- **solar_output** : None, no output_file will be created while a plot will be outputted
- **verbosity_output** : None, no verbosity_output will be created while a plot will be outputted
- **input_start_dt** : None, will use the whole dataset for the computation of the distributions
- **input_end_dt** : None, will use the whole dataset for the computation of the distributions
- **simulation_start_dt** : None, will simulate over the whole dataset
- **simulation_end_dt** : None, will simulate over the whole dataset
- **target_mape** : the mape of the current dataset
- **a** : 4
- **curvature_target** : mean of the second difference of the dataset
- **mip_gap** : 0.3
- **number_simulations** : 1
- **time_limit** : 3600 seconds
- **seed** : 1234
- **verbosity** : 2
- **sid_feature** : "actuals"
- **base_process** : "ARMA"
- **load_pickle** : False
- **curvature** : False
- **show_curv_model** : False
- **solar_plot** : False
- **solver** : gurobi
- **solar_target_scaled_capacity** : None, will not scale scenario data

12.4 Imutable Features

The following MapeMaker options cannot be changed from the command line in `solar_mape_maker`.

- **–scale_by_capacity 0**:

  Scale MAPE by capacity, which is the maximum of the observation data.

- **–target_scaled_capacity None**:

  Simulated data from `MapeMaker` is not scaled, since the input and output of `MapeMaker` are deviations. `--solar_target_scaled_capacity` or `-sts` is used if the user want to scale all scenario data.

CHAPTER 13

Texas_7k_maker

The Texas_7k_maker program enables quick creation of scenario files based on Texas7k data. The options of this program is a subset of the options in Mape_Maker, with additional options for geographic scale (individual sites or sum of all sites) and data source.

13.1 Options

- **–data_source TEXT**:

  The source of simulation data (. . . _test contains smaller datasets for quick test).

  The options to choose from are: “Princeton”, “Princeton_test”, “NREL_ECMWF_PEFORM”, and “NREL_ECMWF_PEFORM_test”.

  The following are the two ways to specify that data source is “Princeton_test”:

  ```
  --data_source "Princeton_test"
  ```

  ```
  -ds "Princeton_test"
  ```

  If the simulation start date is not given, it will use the “Princeton_test” data.

- **–geographic_scale TEXT**:

  Simulation for each individual sites or the sum of all sites.

  Choose “sum” or “individual”.

  The following are the two ways to specify that scenarios are created for each individual sites:

  ```
  --geographic_scale "individual"
  ```

```
-gc "individual"
```

If the geographic_scale is not given, it will create scenarios for the sum of all sites.

- **–output_dir TEXT**:

  Path to destination dir where the scenario are saved as csv file(s).

  The following are the two ways to specify that the output directory is called “output”:

  ```
  --output_dir "output"
  ```

  ```
  -o "output"
  ```

  This option is required if the user choose “sum” as geographic scale, and if this option is not given, a value error will be raised.

  For “individual” as geographic scale, the names of output directory are the names of sites.

- **–simulation_start_dt TEXT**:

  The start date of the simulation of scenarios, must be between the input file date range. (format = “Y-m-d H:M:S”)

  The following are two ways to specify that the start date for the simulation is 2020-1-3 00:00:00 :

  ```
  --simulation_start_dt "2020-1-3 00:00:00"
  ```

  ```
  -ss "2020-1-3 00:00:00"
  ```

  If the simulation start date is not given, it will take the first date of the sid file as simulation start date.

- **–simulation_end_dt TEXT**:

  The end date of the simulation of scenarios, must be between the input file date range. (format = “Y-m-d H:M:S”)

  The following are two ways to specify that the end date for the simulation is 2020-1-3 00:00:00 :

  ```
  --simulation_end_dt "2020-1-3 00:00:00"
  ```

  ```
  -se "2020-1-3 00:00:00"
  ```

  If the simulation end date is not given, it will take the last date of the sid file as simulation end date.

- **–target_mape FLOAT**:

The target MAPE (Mean Absolute Percentage Error) sets the value of the desired MAPE for the simulated data.

The following are the two ways to specify that the target MAPE should be 41.1:

```
--target_mape 41.1
```

```
-t 41.1
```

If this option is not given, the target MAPE is the MAPE of the input data.

- **–number_simulations INTEGER**:

  The number of scenarios to create.

  The following are the two ways to specify that the number of simulations is 4:

  ```
  --number_simulations 4
  ```

  ```
  -n 4
  ```

  If this option is not given, the number of simulations is assumed to be 1.

- **–seed INTEGER**:

  The seed used for simulation. If none, the seed will be random.

  The following are two ways to specify that the title if the seed is set as “1134”:

  ```
  --seed 1134
  ```

  ```
  -s 1134
  ```

  If this option is not given, the seed will be randomly chosen.

- **–t7k_plot BOOLEAN**:

  True if the user wants to plot the results.

  The following are two ways to specify to plot the result:

  ```
  --t7k_plot
  ```

  ```
  -pl
  ```

  If this option is not given, the option is assumed to be True

- **–scale_by_capacity FLOAT**:

Calculate MAPE relative to capacity instead of observations, i.e.

$$mape = \frac{100}{n} \sum_{i=1}^{n} \frac{|f_i - a_i|}{cap}$$

The following are the two ways to specify that the capacity is 2000:

`--scale_by_capacity 2000`

`-sb 2000`

If this option is not given, scale by observation.

If this option is given to be 0, capacity is set to be the maximum of the observation.

- **–target_scaled_capacity FLOAT**:

  Optionally enter target capacity to scale all simulated data by target_capacity/capacity

  The following are the two ways to specify that the target capacity is 1000:

  `--target_scaled_capacity 1000`

  `-ts 1000`

  If this option is not given, simulated data is not scaled.

13.2 Example

```
python -m mape_maker.Texas_7k.Texas_7k_maker -ds "Princeton_test" -gs "individual" -n␣
↪2
```

- **-ds "Princeton_test"**:

  The data source is "Princeton_test".

- **-gs "individual"**:

  The geographic scale is "individual", meaning the scenarios will be created for each individual sites.

- **-n 2**:

  The number of simulations that we want to create is "2". This will create two simulation columns in the output file.

13.3 Imutable Features

The following MapeMaker options cannot be changed from the command line in `Texas_7k_maker`.

- **–sid_feature "actuals"**:

  Set up the the target of the simulation as "actuals". So the `Texas_7k_maker` will simulate the "actuals" data according to the "forecasts" data in the input file.

- **–input_start_dt None**:

Start date for the computation of the distributions is the first date of the input xyid file.

- **–input_end_dt None**:

  End date for the computation of the distributions is the last date of the input xyid file.

- **–time_limit 3600**:

  Time limit for curvature optimization is 3600 seconds.

- **–curvature_target “None”**:

  The target of the second difference is assumed to be the mean of the second difference of the dataset.

- **–verbosity_output “None”**:

  The verbosity output will be shown on terminal.

- **–base_process “ARMA”**:

  Use “ARMA” as the base process. The default base process is set as “ARMA”.

- **–mip_gap “0.3”**:

  Mip gap for curvature optimization is set to 0.3

- **–a 4**:

  When estimating the conditional beta distribution parameters over a sample, 4% of data on the left and 4% on the right is used.

- **–verbosity 2**:

  The verbosity level will set logging.INFO as default (will output info, error, and warning messages).

- **–load_pickle False**:

  The parameters for the beta distributions are computed (no saved pickle file of the estimated parameters).

- **–curvature False**:

  Do not optimize the scenarios curvature.

- **–show_curv_model False**:

  Do not show the model for curvature.

- **–solver “gurobi”**:

  The name of the software that is used to perform the curvature optimization process is “gurobi”.

CHAPTER 14

CAISO_maker

The `CAISO_maker.py` program enables quick creation of scenario files based on wind data obtained from the *CAISO_wind_operational_data.csv* file. The options of this program is a subset of the options in `Mape_Maker`.

14.1 Options

- **–output_dir TEXT**:

  Path to destination dir where the scenario are saved as csv file(s).

  The following are the two ways to specify that the output directory is called “output”:

  `--output_dir "output"`

  `-o "output"`

  This option is required, and if this option is not given, a value error will be raised.

- **–simulation_start_dt TEXT**:

  The start date of the simulation of scenarios, must be between the input file date range. (format = “Y-m-d H:M:S”)

  `--simulation_start_dt "2020-1-3 00:00:00"`

  `-ss "2020-1-3 00:00:00"`

  If the simulation start date is not given, it will take the first date of the sid file as simulation start date (2013-07-01 00:00:00).

- **–simulation_end_dt TEXT**:

  The end date of the simulation of scenarios, must be between the input file date range. (format = "Y-m-d H:M:S")

  ```
  --simulation_end_dt "2020-1-3 00:00:00"
  ```

  ```
  -se "2020-1-3 00:00:00"
  ```

  If the simulation end date is not given, it will take the last date of the sid file as simulation end date (2015-06-30 23:00:00).

- **–target_mape FLOAT**:

  The target MAPE (Mean Absolute Percentage Error) sets the value of the desired MAPE for the simulated data. The MAPE will be computed based on capacity. The following are the two ways to specify that the target MAPE should be 41.1:

  ```
  --target_mape 41.1
  ```

  ```
  -t 41.1
  ```

  If this option is not given, the target MAPE is the MAPE of the input data.

- **–number_simulations INTEGER**:

  The number of scenarios to create.

  The following are the two ways to specify that the number of simulations is 4:

  ```
  --number_simulations 4
  ```

  ```
  -n 4
  ```

  If this option is not given, the number of simulations is assumed to be 1.

- **–seed INTEGER**:

  The seed used for simulation.

  The following are two ways to specify that the title if the seed is set as "1134":

  ```
  --seed 1134
  ```

  ```
  -s 1134
  ```

  In this option is not given, the seed used for simulation is 1234.

- **–plot BOOLEAN**:

  True if the user wants to plot the results.

  The following are two ways to specify to plot the result:

  `--plot`

  `-p`

  If this option is not given, the option is assumed to be False

14.2 Example

```
python -m mape_maker.CAISO_maker -o "CAISO_maker_test_output" -n 3 -ss "2013-07-01␣
↪00:00:00" -se "2014-07-01 00:00:00" -p
```

- **-o “CAISO_maker_test_output”**:

  Create an output directory called “CAISO_maker_test_output”, in which will store the simulation output file.

- **-n 3**:

  The number of simulations that we want to create is “3”. This will create three simulation columns in the output file.

- **-ss “2013-07-01 00:00:00”**:

  The start time of the simulation is “2013-07-01 00:00:00”.

- **-se “2014-07-01 00:00:00”**:

  The end time of the simulation is “2014-07-01 00:00:00”.

- **-p**:

  Plot the output

14.3 Imutable Features

The following MapeMaker options cannot be changed from the command line in `CAISO_maker`.

- **–input_sid_file “mape_maker/samples/CAISO_wind_operational_data.csv”**:

  The csv file containing CAISO data.

- **–sid_feature “actuals”**:

  Set up the the target of the simulation as “actuals”. So the `CAISO_maker` will simulate the “actuals” data according to the “forecasts” data in the input file.

- **–input_start_dt None**:

  Start date for the computation of the distributions is the first date of the input xyid file.

- **–input_end_dt None**:

End date for the computation of the distributions is the last date of the input xyid file.

- **–time_limit 3600**:

  Time limit for curvature optimization is 3600 seconds.

- **–curvature_target "None"**:

  The target of the second difference is assumed to be the mean of the second difference of the dataset.

- **–verbosity_output "None"**:

  The verbosity output will be shown on terminal.

- **–base_process "ARMA"**:

  Use "ARMA" as the base process. The default base process is set as "ARMA".

- **–mip_gap "0.3"**:

  Mip gap for curvature optimization is set to 0.3

- **–a 4**:

  When estimating the conditional beta distribution parameters over a sample, 4% of data on the left and 4% on the right is used.

- **–verbosity 2**:

  The verbosity level will set logging.INFO as default (will output info, error, and warning messages).

- **–load_pickle False**:

  The parameters for the beta distributions are computed (no saved pickle file of the estimated parameters).

- **–curvature False**:

  Do not optimize the scenarios curvature.

- **–show_curv_model False**:

  Do not show the model for curvature.

- **–solver "gurobi"**:

  The name of the software that is used to perform the curvature optimization process is "gurobi".

- **–scale_by_capacity 0**:

  Scale by capacity, which is the maximum of the observation data.

- **–target_scaled_capacity None**:

  Simulated data is not scaled.

CHAPTER 15

BPA_maker

The `BPA_maker.py` program enables quick creation of scenario files based on wind data obtained from the *2012-2013_BPA_forecasts_actuals.csv* file. The options of this program is a subset of the options in `Mape_Maker`.

15.1 Options

- **–output_dir TEXT**:

  Path to destination dir where the scenario are saved as csv file(s).

  The following are the two ways to specify that the output directory is called “output”:

  `--output_dir "output"`

  `-o "output"`

  This option is required, and if this option is not given, a value error will be raised.

- **–simulation_start_dt TEXT**:

  The start date of the simulation of scenarios, must be between the input file date range. (format = “Y-m-d H:M:S”)

  `--simulation_start_dt "2020-1-3 00:00:00"`

  `-ss "2020-1-3 00:00:00"`

  If the simulation start date is not given, it will take the first date of the sid file as simulation start date (2012-06-02 00:00:00).

- **–simulation_end_dt TEXT**:

  The end date of the simulation of scenarios, must be between the input file date range. (format = “Y-m-d H:M:S”)

  ```
  --simulation_end_dt "2020-1-3 00:00:00"
  ```

  ```
  -se "2020-1-3 00:00:00"
  ```

  If the simulation end date is not given, it will take the last date of the sid file as simulation end date (2014-01-01 23:00:00).

- **–target_mape FLOAT**:

  The target MAPE (Mean Absolute Percentage Error) sets the value of the desired MAPE for the simulated data. The MAPE will be computed based on capacity. The following are the two ways to specify that the target MAPE should be 41.1:

  ```
  --target_mape 41.1
  ```

  ```
  -t 41.1
  ```

  If this option is not given, the target MAPE is the MAPE of the input data.

- **–number_simulations INTEGER**:

  The number of scenarios to create.

  The following are the two ways to specify that the number of simulations is 4:

  ```
  --number_simulations 4
  ```

  ```
  -n 4
  ```

  If this option is not given, the number of simulations is assumed to be 1.

- **–seed INTEGER**:

  The seed used for simulation.

  The following are two ways to specify that the title if the seed is set as “1134”:

  ```
  --seed 1134
  ```

  ```
  -s 1134
  ```

  In this option is not given, the seed used for simulation is 1234.

- **–plot BOOLEAN**:

  True if the user wants to plot the results.

  The following are two ways to specify to plot the result:

  `--plot`

  `-p`

  If this option is not given, the option is assumed to be False

15.2 Example

```
python -m mape_maker.BPA_maker -o "BPA_maker_test_output" -n 3 -ss "2013-01-01␣
↪00:00:00" -se "2013-07-01 00:00:00" -p
```

- **-o "BPA_maker_test_output"**:

  Create an output directory called "BPA_maker_test_output", in which will store the simulation output file.

- **-n 3**:

  The number of simulations that we want to create is "3". This will create three simulation columns in the output file.

- **-ss "2013-01-01 00:00:00"**:

  The start time of the simulation is "2013-01-01 00:00:00".

- **-se "2013-07-01 00:00:00"**:

  The end time of the simulation is "2013-07-01 00:00:00".

- **-p**:

  Plot the output

15.3 Imutable Features

The following MapeMaker options cannot be changed from the command line in `BPA_maker`.

- **–input_sid_file "mape_maker/samples/2012-2013_BPA_forecasts_actuals.csv"**:

  The csv file containing BPA data.

- **–sid_feature "actuals"**:

  Set up the the target of the simulation as "actuals". So the `BPA_maker` will simulate the "actuals" data according to the "forecasts" data in the input file.

- **–input_start_dt None**:

  Start date for the computation of the distributions is the first date of the input xyid file.

- **–input_end_dt None**:

End date for the computation of the distributions is the last date of the input xyid file.

- **–time_limit 3600**:

  Time limit for curvature optimization is 3600 seconds.

- **–curvature_target "None"**:

  The target of the second difference is assumed to be the mean of the second difference of the dataset.

- **–verbosity_output "None"**:

  The verbosity output will be shown on terminal.

- **–base_process "ARMA"**:

  Use "ARMA" as the base process. The default base process is set as "ARMA".

- **–mip_gap "0.3"**:

  Mip gap for curvature optimization is set to 0.3

- **–a 4**:

  When estimating the conditional beta distribution parameters over a sample, 4% of data on the left and 4% on the right is used.

- **–verbosity 2**:

  The verbosity level will set logging.INFO as default (will output info, error, and warning messages).

- **–load_pickle False**:

  The parameters for the beta distributions are computed (no saved pickle file of the estimated parameters).

- **–curvature False**:

  Do not optimize the scenarios curvature.

- **–show_curv_model False**:

  Do not show the model for curvature.

- **–solver "gurobi"**:

  The name of the software that is used to perform the curvature optimization process is "gurobi".

- **–scale_by_capacity 0**:

  Scale by capacity, which is the maximum of the observation data.

- **–target_scaled_capacity None**:

  Simulated data is not scaled.

CHAPTER 16

fake_BPA_maker

The `fake_BPA_maker.py` program enables quick creation of scenario files based on wind data obtained from the *fake_bpa_data.csv* file. The options of this program is a subset of the options in `Mape_Maker`.

16.1 Options

- **–output_dir TEXT**:

  Path to destination dir where the scenario are saved as csv file(s).

  The following are the two ways to specify that the output directory is called “output”:

  `--output_dir "output"`

  `-o "output"`

  This option is required, and if this option is not given, a value error will be raised.

- **–simulation_start_dt TEXT**:

  The start date of the simulation of scenarios, must be between the input file date range. (format = “Y-m-d H:M:S”)

  `--simulation_start_dt "2020-1-3 00:00:00"`

  `-ss "2020-1-3 00:00:00"`

  If the simulation start date is not given, it will take the first date of the sid file as simulation start date (2012-06-02 00:00:00).

- **–simulation_end_dt TEXT**:

  The end date of the simulation of scenarios, must be between the input file date range. (format = “Y-m-d H:M:S”)

  ```
  --simulation_end_dt "2020-1-3 00:00:00"
  ```

  ```
  -se "2020-1-3 00:00:00"
  ```

  If the simulation end date is not given, it will take the last date of the sid file as simulation end date (2014-01-01 23:00:00).

- **–target_mape FLOAT**:

  The target MAPE (Mean Absolute Percentage Error) sets the value of the desired MAPE for the simulated data. The MAPE will be computed based on capacity. The following are the two ways to specify that the target MAPE should be 41.1:

  ```
  --target_mape 41.1
  ```

  ```
  -t 41.1
  ```

  If this option is not given, the target MAPE is the MAPE of the input data.

- **–number_simulations INTEGER**:

  The number of scenarios to create.

  The following are the two ways to specify that the number of simulations is 4:

  ```
  --number_simulations 4
  ```

  ```
  -n 4
  ```

  If this option is not given, the number of simulations is assumed to be 1.

- **–seed INTEGER**:

  The seed used for simulation.

  The following are two ways to specify that the title if the seed is set as “1134”:

  ```
  --seed 1134
  ```

  ```
  -s 1134
  ```

  In this option is not given, the seed used for simulation is 1234.

- **–plot BOOLEAN**:

  True if the user wants to plot the results.

  The following are two ways to specify to plot the result:

  `--plot`

  `-p`

  If this option is not given, the option is assumed to be False

16.2 Example

```
python -m mape_maker.fake_BPA_maker -o "fake_BPA_maker_test_output" -n 3 -ss "2013-01-01 00:00:00" -se "2013-07-01 00:00:00" -p
```

- **-o "fake_BPA_maker_test_output"**:

  Create an output directory called "fake_BPA_maker_test_output", in which will store the simulation output file.

- **-n 3**:

  The number of simulations that we want to create is "3". This will create three simulation columns in the output file.

- **-ss "2013-01-01 00:00:00"**:

  The start time of the simulation is "2013-01-01 00:00:00".

- **-se "2013-07-01 00:00:00"**:

  The end time of the simulation is "2013-07-01 00:00:00".

- **-p**:

  Plot the output

16.3 Imutable Features

The following MapeMaker options cannot be changed from the command line in `fake_BPA_maker`.

- **–input_sid_file "mape_maker/samples/fake_bpa_data.csv"**:

  The csv file containing fake BPA data.

- **–sid_feature "actuals"**:

  Set up the the target of the simulation as "actuals". So the `fake_BPA_maker` will simulate the "actuals" data according to the "forecasts" data in the input file.

- **–input_start_dt None**:

  Start date for the computation of the distributions is the first date of the input xyid file.

- **–input_end_dt None**:

End date for the computation of the distributions is the last date of the input xyid file.

- **–time_limit 3600**:

  Time limit for curvature optimization is 3600 seconds.

- **–curvature_target “None”**:

  The target of the second difference is assumed to be the mean of the second difference of the dataset.

- **–verbosity_output “None”**:

  The verbosity output will be shown on terminal.

- **–base_process “ARMA”**:

  Use “ARMA” as the base process. The default base process is set as “ARMA”.

- **–mip_gap “0.3”**:

  Mip gap for curvature optimization is set to 0.3

- **–a 4**:

  When estimating the conditional beta distribution parameters over a sample, 4% of data on the left and 4% on the right is used.

- **–verbosity 2**:

  The verbosity level will set logging.INFO as default (will output info, error, and warning messages).

- **–load_pickle False**:

  The parameters for the beta distributions are computed (no saved pickle file of the estimated parameters).

- **–curvature False**:

  Do not optimize the scenarios curvature.

- **–show_curv_model False**:

  Do not show the model for curvature.

- **–solver “gurobi”**:

  The name of the software that is used to perform the curvature optimization process is “gurobi”.

- **–scale_by_capacity 0**:

  Scale by capacity, which is the maximum of the observation data.

- **–target_scaled_capacity None**:

  Simulated data is not scaled.