SAND2019-10500R

Mape_Maker: A Scenario Creator

Guillaume Goujard – University of California Berkeley
Jean-Paul Watson – Sandia National Laboratories
David L. Woodruff – University of California Davis

September 5, 2019

Abstract

We describe algorithms for creating probabilistic scenarios for the situation when the underlying forecast methodology is modeled as being more (or less) accurate than it has been historically. Such scenarios can be used in studies that extend into the future and may need to consider the possibility that forecast technology will improve. Our approach can also be used to generate alternative realizations of renewable energy production that are consistent with historical forecast accuracy, in effect serving as a method for creating families of realistic alternatives – which are often critical in simulation-based analysis methodologies.

1 Introduction

Uncertainty associated with the forecasted output of renewable energy sources such as wind and solar mandates analysis and management techniques that take stochastics into account. A growing literature describes methods for creating and evaluating probabilistic *scenarios*, which are forecasts of renewables power generation with an attached probability. A representative sample of this literature can be found in [2, 3, 4, 5, 6, 8]. Here, we are interested in creating probabilistic scenarios for the situation when the underlying forecast methodology is modeled as being more (or less) accurate than it has been historically. Such scenarios can be used in studies that extend into the future and may need to consider the possibility that forecast technology will improve. Our approach can also be used to generate alternative realizations of renewable energy production that are consistent with historical forecast accuracy, in effect serving as a method for creating families of realistic alternatives – which are often critical in simulation-based analysis methodologies. A general open-source software implementation of the methods described here – a package called *mape_maker* – is publicly available at `https://github.com/mape-maker/mape-maker`.

Given a time series of forecasts (e.g., daily over a year), we create a set of scenarios for renewable power production that, based on a forecast system with a specified accuracy, could reasonably correspond to the forecasts. We often refer to these scenarios as *actuals*, to distinguish these values from historical forecasts. We can also create a set of forecasts that could reasonably correspond to a given time series of actuals. In other words, the process can be inverted. The correspondence between forecasts and actuals is based on analysis of historic forecast error distributions. Subsequently, the word "reasonably" is replaced with mathematical criteria concerning the error distribution, temporal correlation, and in the case of the forecast, curvature. As a preview of the output of

this capability, consider Figure 1. This figure provides a simple example where a set of 5 alternative "actual" scenarios are constructed for a few days in July of 2013 based on wind forecast error data from obtained from the California Independent System Operator (CAISO) in the US for July 2013 through May 2015. The target error – specifically, the mean absolute percentage error or MAPE – is the value that was realized in the forecast error data. Because the scenarios are created for days in the past, we are able so show both the forecast and realized actuals on the same plot as the constructed scenarios.

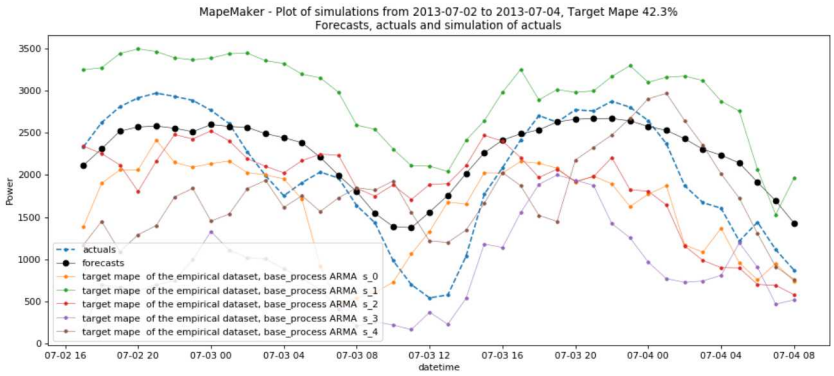


Figure 1: Illustration of 5 scenarios of wind production in CAISO representing alternative actuals. The forecast and realized actuals are also shown.

1.1 Measures of Forecast Error

Let $(x_i)_i \in \mathbb{R}^n$ and $(y_i)_i \in \mathbb{R}^n$ denote two time-series. For simplicity, we subsequently refer to these time-series as x and y. We then define the following functions:

$$\begin{array}{rccccl} RE & : & \mathbb{R}^* \times \mathbb{R} & \to & \mathbb{R} & \text{(Relative Error)} \\ & & x, y & \mapsto & \frac{y-x}{x} & \end{array}$$

$$\begin{array}{rccccl} MARE & : & \mathbb{R}^{*n} \times \mathbb{R}^n & \to & \mathbb{R}_+ & \text{(Mean Absolute Relative Error)} \\ & & x, y & \mapsto & \sum_{i=1}^n \frac{|RE(x_i,y_i)|}{n} & \end{array}$$

The MAPE (Mean Absolute Percentage Error) is simply the MARE (Mean Absolute Relative Error) given as a percentage. Our software library communicates with users in terms of MAPE, but in our discussions here it is convenient to use MARE and sometimes MAE (Mean Absolute Error) variants.

While MAPE is a very popular way of characterizing forecast accuracy for renewables production, it is well-known to have a number of undesirable properties (see, e.g., [7]). One undesirable properties is that x values of zero must be ignored in the calculation. We have organized our methods in such a way as to avoid division by zero. Most of the development here is based on converting the MAPE target to an absolute error conditional on the value of x, so it would be relatively straightforward extension to convert our algorithms to use some measure of accuracy other than the MAPE.

1.2 Notation Scheme

We use $\mathcal{X}$ and $\mathcal{Y}$ to denote paired input data of length n. Note that which of these pairs is the forecast and which is the actual depends on the user objective, i.e., what is being

simulated. For example, if one desires to obtain alternative actuals from forecasts, then $\mathcal{Y}$ will be simulated actuals. Recalling the canonical goal of constructing a vector Y from X input for a specified range of dates, we use $\mathcal{X}_{SID}$ to denote the input data upon which the construction is based (Simulation Input Data). It may or may not be the case that $\mathcal{X}_{SID}$ is a subset of $\mathcal{X}$. For the next few sections, we assume that both input datasets are sorted according to the X values, e.g., $\mathcal{X} = (x_i)_{i \leq n}$ with n equal to the cardinality of $\mathcal{X}$ such that $\forall i < j$, $x_i < x_j$. We will return to a temporal sorting in Section 3.5 when we consider auto-correlation.

We use bold upper case font to denote random variables. As indicated above, the role of the forecasts and the actuals can be reversed. If we want to compute y, then x is the input data for the simulation. We let $\mathcal{E}$ denote a random vector of errors such that $\mathcal{E}_i = \mathbf{Y}_i - x_i$ so

$$\mathbf{Y}_i = x_i + \mathcal{E}_i, \qquad \forall i \leq n.$$

We let ε denote a vector of observed errors. We will focus on the modeling of ε in the following. The title of the paper and the name of our software library derives from the requirement that simulated values $\tilde{y}$ must result in a MAPE close enough to the target MAPE. We formalize this constraint as

$$\mathbb{E}[MARE(x, \tilde{\mathbf{Y}})] = \tilde{r},$$

where $\tilde{r}$ is the target MAPE divided by 100%.

1.3 Plausibility Criteria

A main theme underlying this work that we will use to justify some of our design choices involves what we refer to as *plausibility criteria*. For any requested MAPE, the distributions of errors computed should be as close as possible to the original error distributions while satisfying the target MAPE. If a user were to select the estimated MAPE as the requested one, one would naturally expect the distribution of errors drawn from the simulated distributions to be somehow "close" to the estimated distribution. For example, if the system of forecasts is producing a wide range of errors at very low forecasted power output, then even if the forecast technology is improving one would expect it to still produce a relatively wider range of errors at low power regardless of the requested MAPE. We formalize these criteria as follows in Definition 1.1.

Definition 1.1. A scenario set is said to be *plausible* if:

1. The error distribution for the set is close to the empirical distribution of errors, i.e, its plausibility score is close to 1 (as defined in later in Section 3.4);

2. the computed auto-correlation coefficients for the set are close the empirical values; and

3. the computed curvature for the set is close to the empirical value, especially when the scenarios are forecasts (because we observe that forecasts typically have lower curvature than actuals.)

2 Modeling the Joint Distribution of $(\mathcal{E}, \mathbf{X})$

Let us define $\mathbf{Z} = (\mathcal{E}, \mathbf{X})$. Here, $\mathbf{Z}$ denotes a random variable with values in $(-\infty, +\infty) \times (0, +\infty)$ – or, if the production capacity *cap* is known by the forecaster, values in $[-cap, cap] \times [0, cap]$. We denote by $f_{\mathbf{Z}}$ the density of $\mathbf{Z}$, and denote by $f_{\mathcal{E}}$ and $f_{\mathbf{X}}$ the marginals of $f_{\mathbf{Z}}$. Then,

$$f_{\mathcal{E}}(\varepsilon) = \int_{-\infty}^{\infty} f_{\mathbf{Z}}(\varepsilon, x)dx, \qquad f_{\mathbf{X}}(x) = \int_{-\infty}^{\infty} f_{\mathbf{Z}}(\varepsilon, x)d\varepsilon$$

We also define the conditional density of $\mathcal{E}$ given $\mathbf{X} = x$ as:

$$f_{\mathcal{E}|\mathbf{X}=x}(\varepsilon) = \frac{f(\varepsilon, x)}{f_{\mathbf{X}}(x)}$$

Modeling the conditional distribution of errors is important as these distributions can vary significantly with the value of input data. For example, when the forecasts and the actuals are both low, the errors will be biased because the power cannot be below zero. Symmetrically, close to the maximum capacity, *cap*, errors are bounded by the fact that power cannot exceed maximum production capacity.

In this context, we introduce the functional $m(x)$ to denote the expected value of the absolute error of the distribution conditioned on x, defined as:

$$m(x) = \mathbb{E}[|\mathcal{E}| \; | \mathbf{X} = x] = \int_{\varepsilon=-\infty}^{\infty} |\varepsilon| f_{\mathcal{E}|\mathbf{X}=x}(\varepsilon)d\varepsilon$$

We then introduce r to denote the mean absolute relative error, defined as:

$$r = \mathbb{E}[\mathbb{E}[\frac{|\mathcal{E}| \; | \mathbf{X}}{\mathbf{X}}]] = \mathbb{E}[\frac{m(\mathbf{X})}{\mathbf{X}}]$$

In Figure 2, we provide an illustrative visualization of the relative error *RE* as a function of actuals. We note that because actuals are correlated with forecasts, the figure would be very similar if forecasts were used in instead. The data is for CAISO wind power data, ranging from July 1, 2013 to June 30, 2015. We will use this dataset for illustration throughout the paper, and refer to it informally as the *CAISO Wind* data set. These data are available in the mape_maker software distribution; the file is `wind_total_forecast_actual_070113_063015.csv`.

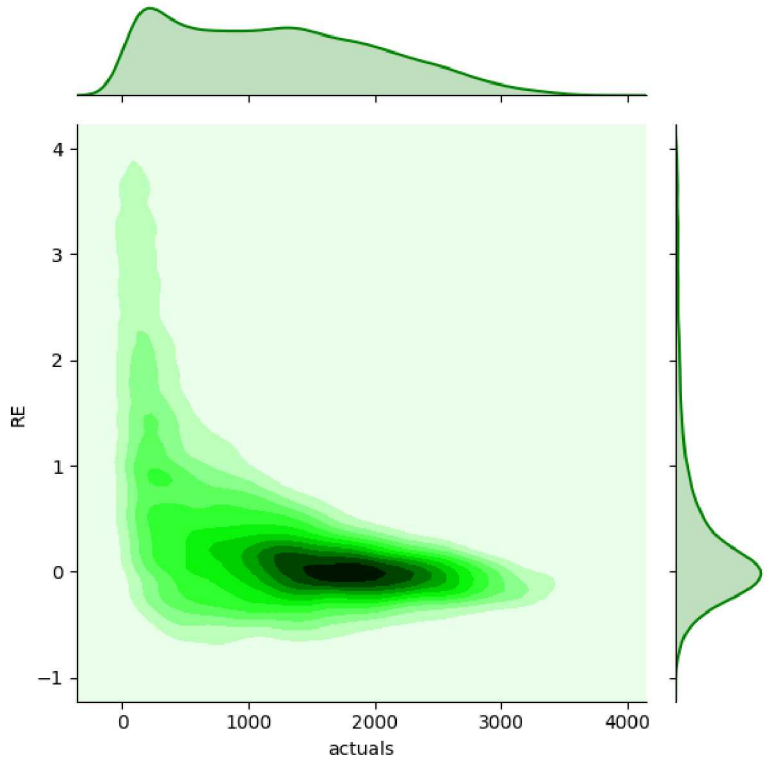


Figure 2: Empirical joint distribution of $(\frac{\mathcal{E}}{\mathbf{X}}, \mathbf{X})$ - CAISO Wind Power

2.1 Estimating the Conditional Distribution of $\mathcal{E}|\mathbf{X}$, $\hat{f}_{\mathcal{E}|\mathbf{X}=x}$

Given the notation $x \in \mathcal{X}$, we use the beta distribution on $[l, s+l]$ to model $f_{\mathcal{E}|X=x}$. In addition to the l and s that we will refer to location parameters, a beta distribution requires two additional parameters – α and β, i.e., the shape parameters. We chose the beta distribution because it has finite support that helps us avoid power values below zero or above *cap* and because the shape parameters provide the flexibility necessary to model different behaviors for each x. We then define

$$\boxed{f_{\mathcal{E}|\mathbf{X}=x}(\varepsilon) = beta(\varepsilon;\ (\alpha, \beta, l, s)) = \frac{(\frac{\varepsilon - l}{s})^{\alpha-1}(1 - \frac{\varepsilon - l}{s})^{\beta-1}}{B(\alpha, \beta)}}$$

with

$$B(\alpha, \beta) = \int_{\varepsilon=l}^{l+s} (\frac{\varepsilon - l}{s})^{\alpha-1}(1 - \frac{\varepsilon - l}{s})^{\beta-1} d\varepsilon$$

2.2 Intervals for Conditional Estimation

We now define a rule that will be used to estimate the parameters of the conditional density based on each x of the input dataset. We choose to take a fraction a (e.g., 0.05) of data before and after each x. Let G_X denote the empirical cumulative distribution function. Then, let $I_x^a = [G_X^{-1}(G_X(x) - a),\ G_X^{-1}(G_X(x) + a)]$. Thus I_x^a is centered on $\bar{x}(x;\ a) = \frac{G_X^{-1}(G_X(x)-a) + G_X^{-1}(G_X(x)+a)}{2}$ with $2a$ fraction of the data. We fit the parameters on the sample $E_{I_x^a} = \{\varepsilon_i,\ 1 \leq i \leq n, x_i \in I_x^a, \}$. Note that for production values near zero and near the capacity, there could be as few as a fraction of the values used.

To compute the estimation for a particular value x', our method uses the interval I_x^a for which $\bar{x}(x;a)$ is closest to x' and uses the corresponding set $E_{I_x^a}$ to compute the parameters for x'. For $x' \in \mathcal{X}$ that are not close to zero or *cap*, the closest $\bar{x}(x;a)$ to x' will often be just $\bar{x}(x';a)$. However, for very small or large values of x' and when $\mathcal{X}_{SID} \not\subset \mathcal{X}$, the use of the interval with the closest mean is most appropriate.

We will now describe how our method fits the parameters of the beta distributions. Because every estimated quantity will depend on a, we drop a as a subscript or function parameter for notational simplicity.

2.3 Fixing l, s and Estimating α, β

2.3.1 Constraints on the Location Parameters

An informed choice of the location parameters will avoid simulating errors leading to y values lower than 0 or greater then the *cap* of the dataset. We now define the function y_{max}, which returns the maximum possible simulated value at x according to a conditional distribution $f_{\mathcal{E}|\mathbf{X}=x}$. Because the inverse of the corresponding cumulative distribution function (CDF) evaluated at one, $F^{-1}_{\mathcal{E}|\mathbf{X}=x}(1) = l + s$, is the maximum of the error simulated; $F^{-1}_{\mathcal{E}|\mathbf{X}=x}(0) = l$ is the minimum; and because we want to avoid simulating values above the cap or below zero we have

$$\begin{aligned} y_{max}(x) &= x + F^{-1}_{\mathcal{E}|\mathbf{X}=x}(1) \\ &= x + s + l \\ &\leq cap. \end{aligned}$$

Similarly,

$$\begin{aligned} y_{min}(x) &= x + F^{-1}_{\mathcal{E}|\mathbf{X}=x}(0) \\ &= x + l \\ &\geq 0. \end{aligned}$$

These two conditions give

$$\begin{aligned} l &\geq -x \\ s &\leq cap - x - l. \end{aligned}$$

Thus, we can define the estimators of the location parameters for each x as:

$$\hat{l}(x) = \begin{cases} -x & \text{if } min\big(\varepsilon_i,\ x_i \in I_x\big) \leq -x \\ min\big(\varepsilon_i,\ x_i \in I_x\big) & \text{else} \end{cases}$$

$$\hat{s}(x) = \begin{cases} cap - x - \hat{l}(x) & \text{if } max\big(\varepsilon_i,\ x_i \in I_x\big) \geq cap - x \\ max\big(\varepsilon_i,\ x_i \in I_x\big) - \hat{l}(x) & \text{else} \end{cases}$$

2.3.2 Choosing the Shape Parameters by the Method of Moments

The mean and variance of a beta(α, β, l, s) distribution are:

$$\mu = \frac{s\alpha}{\beta + \alpha} + l$$
$$V = \frac{1}{s^2} \frac{\alpha\beta}{(\alpha + \beta)^2(\alpha + \beta + 1)}$$

We can now choose shape parameters by solving these two equations for α and β

$$\hat{\mu}(x) = \frac{\hat{s}(x)\alpha}{\beta + \alpha} + \hat{l}(s)$$
$$\hat{V}(x) = \frac{1}{\hat{s}(x)^2} \frac{\alpha\beta}{(\alpha + \beta)^2(\alpha + \beta + 1)}$$

to obtain $\hat{\alpha}(x)$ and $\hat{\beta}(x)$.

For any $x \in \mathcal{X} \cup \mathcal{X}_{SID}$ assign

$$\hat{\mathcal{S}}_x = (\hat{\alpha}(x), \hat{\beta}(x), \hat{l}(x), \hat{s}(x))$$

2.4 Selecting a

We now develop an empirical way to select the best a. If a is small, the sample on which to fit the distribution will be small since I_x^a is small. Fitting a distribution on very little data is of course dangerous. On the other hand, if a is large, then the sample is too large to provide us with an estimation of the conditional density. In the extreme where $a = 1$, every conditional density will be equal to the density of the relative error.

One way to select a is to compute a discrepancy score between the empirical distribution function and the one obtained by estimating each conditional distributions with $2a$ of the data. Let g be the empirical joint density of (X, ε). Let $\hat{f}$ be the joint density of $(\mathbf{X}, \mathcal{E})$ taken as $\hat{f}_a(x, \varepsilon) = \hat{f}_{\mathbf{X}}(x) * \hat{f}^a_{\mathcal{E}|\mathbf{X}=x}(\varepsilon)$. We choose a to minimize the deviation between the real density and the simulated density:

$$D^2(a) = \int_x \int_\varepsilon (g(x, \varepsilon) - \hat{f}_a(x, \varepsilon))^2 d\varepsilon dx.$$

3 Adjusting the conditional densities to fit a MARE target

We will use a tilde to specify the distributions and variables that we are simulating.

- While $\mathcal{E}$ is the random variable of the error with properties that can be estimated from $\mathcal{X}$, $\tilde{\mathcal{E}}$ is the random variable of error defined by a distribution that we will develop with desired properties for the simulation.
- We make use of three conditional distributions : the population density, $f_{\mathcal{E}|\mathbf{X}=x}$, the estimated density $\hat{f}_{\mathcal{E}|\mathbf{X}=x}$, and a simulation density, $\tilde{f}_{\mathcal{E}|\mathbf{X}=x}$.

We are now interested in modeling the conditional distribution of $\tilde{\mathcal{E}}|\mathbf{X}$ so that the expected relative absolute error of the simulated random variable $\tilde{\mathcal{E}}$ is :

$$\mathbb{E}_{\tilde{\mathcal{E}}}[\frac{1}{n_{SID}}\sum_{x\in\mathcal{X}_{SID}}\frac{|\tilde{\mathcal{E}}|\ |\mathbf{X}=x}{x}] = \tilde{r}$$

3.1 Adjusting the shape parameters so that it fits a target MAE

We want to adjust each conditional distribution so that the global distribution of $\tilde{\mathcal{E}}$ satisfies the targeted MARE and so that they keep the same shape parameters as the original distributions. To do this we compute analytically the mean absolute error of a beta distribution when α and β are fixed. Let $l < 0$ and $s + l > 0$. Let $b(\cdot; \alpha, \beta, l, s)$ be an arbitrary beta density function with parameters (α, β, l, s) for which we define a mean absolute error function of l and s given values for α and β as

$$\nu(l, s; \alpha, \beta) = \int_{\varepsilon=l}^{s+l} |\varepsilon| b(\varepsilon; \alpha, \beta, l, s) d\varepsilon.$$

We will make two remarks:

$$\lim_{s\to 0} \nu(l, s; \alpha, \beta) = 0, \quad \forall l < 0$$

$$\nu(l, s; \alpha, \beta) \underset{s\to\infty}{\sim} \frac{s\alpha}{\alpha + \beta}$$

Since ν is continuous (it is a sum of continuous functions), the intermediate value theorem applies which means that $\nu(l, s; \alpha, \beta)$ can achieve any value and in particular, the value needed to in order to hit the specified error target.

Thus, once we are given α, β, and a target value for the absolute error at a particular value of x, we need to find the intersection between a hyperplane defined by the target and the surface defined by $\nu(l, s; \alpha, \beta)$ to establish values for $\tilde{l}$ and $\tilde{s}$. For $x \in \mathbb{R}_+$ we will want to choose the solution that minimize the distance to the estimated values $\hat{l}(x)$ and $\hat{s}(x)$ while hitting a target mean absolute error $m(x)$ and without changing the shape parameters.

$$\begin{aligned}(\tilde{l}(x), \tilde{s}(x)) = \underset{l,s}{\text{argmin}} \quad & (l - \hat{l}(x))^2 + (s - \hat{s}(x))^2 \\ \text{s.t.} \quad & l \in \mathbb{R},\ s \in \mathbb{R}_+ \\ & 0 \geq l \geq -x \\ & 0 \leq s \leq cap - x - l \\ & \nu(l, s; \hat{\alpha}(x), \hat{\beta}(x)) = m(x)\end{aligned} \tag{1}$$

However, in our case, since there are bound constraints on l and s (see section 2.3.1), ν cannot hit every target $m(x)$. We compute a maximum target function that can be hit as:

$$m_{max}(x) = \max_{l\in(-x,0], s\in[0, cap-x)} \nu(l, s; \hat{\alpha}(x), \hat{\beta}(x))$$

The target function m must then be bounded for every x by :

$$m(x) \leq m_{max}(x) \tag{2}$$

Given a mean absolute error target function m satisfying inequality (2) we obtain for any x, a beta distribution of parameters $\tilde{\mathcal{S}}_{x,m} = (\hat{\alpha}(x), \hat{\beta}(x), \tilde{l}(x), \tilde{s}(x))$ that satisfies the mean absolute error target and that is the closest possible to the estimated distribution. We now proceed to allocate an error target to each $x \in \mathcal{X}_{SID}$ that we will call $\tilde{m}$ that depends on the target MARE and on a weight function.

3.2 Changing the conditional distributions

3.2.1 Weight functions

Let's define $\Omega_{\mathcal{X}_{SID}}$ as the set of functions $\omega_{\mathcal{X}_{SID}}$ defined on $\mathcal{X}_{SID}$ such that

$$\frac{1}{n_{SID}} \sum_{x \in \mathcal{X}_{SID}} \omega_{\mathcal{X}_{SID}}(x) = 1.$$

We call them *weight functions*. Weight functions will be used to assign a target MAE to obtain from each of the conditional distributions $\tilde{\mathcal{E}}|\mathbf{X} = x$, for all $x \in \mathcal{X}_{SID}$. It can also be seen as the function that weights the contribution of the Absolute Error of each conditional distribution to the Mean Absolute Relative Error of the simulation.

3.2.2 Target function generator

We also define the following functional that we call target function generator.

$$\begin{array}{rccll} \tilde{m} & : & \mathcal{X}_{SID} \times \mathbb{R}_+ \times \Omega_{\mathcal{X}_{SID}} & \to & \mathbb{R}_+ \qquad \text{(Target function generator)} \\ & & x, \tilde{r}, \omega & \mapsto & \tilde{r} x \omega(x),\ x > 0 \end{array}$$

For a fixed $\tilde{r}$ and ω, $\tilde{m}(.\ , \tilde{r}, \omega)$ is a target function. Since the target function will be used to directly adjust the conditional distribution, it must respect the inequality (2). Finally, we say that a target mare $\tilde{r}$ is *feasible* for a given $\omega \in \Omega_{\mathcal{X}_{SID}}$ if

$$\forall x \in \mathcal{X},\ \tilde{m}(x, \tilde{r}, \omega) \leq m_{max}(x)$$

3.2.3 Zero power input

We recall that the zero input does not count in the computation of the MARE. However, we want the distribution of the simulated errors to be drawn from the estimated distribution. In other words :

$$\forall \tilde{r} \in \mathbb{R}_+, \quad \tilde{l}(0) = \hat{l}(0) \ \textit{and}\ \tilde{s}(0) = \hat{s}(0)$$

We assign

$$\tilde{m}(0) = \hat{m}(0)$$

To avoid big discontinuities in the parameters of the beta distributions, we could take as $\tilde{l}(0) = \lim_{x \to 0} \tilde{\ell}(x)$, $\tilde{s}(0) = \lim_{x \to 0} \tilde{s}(x)$,

3.2.4 Convergence to the requested MARE

Using the function $\tilde{m}$ to assign target MAE for each SID input will allow us to hit the targeted MARE using the simulation distribution. Indeed, let us define the random variable $\tilde{\mathcal{E}}|\mathbf{X}$ with density $\tilde{f}_{\mathcal{E}|\mathbf{X}=x}(\varepsilon) = b(\varepsilon, \tilde{\mathcal{S}}_{x,\tilde{m}})$, $\varepsilon \in (-cap, cap)$. If we establish the distribution parameters as described in Section 3.1 and solve program (1) with $m(x) = \tilde{m}(x;\, \tilde{r}, \omega_{\mathcal{X}_{SID}})$ we have,

$$\int_{\varepsilon=-\infty}^{\infty} |\varepsilon| b(\varepsilon, \tilde{\mathcal{S}}_{x,\tilde{m}}) d\varepsilon = \tilde{m}(x;\, \tilde{r}, \omega_{\mathcal{X}_{SID}}), \quad \forall x \in \mathcal{X}_{SID}.$$

Then, the expected MARE with the errors drawn from these distributions and with the inputs in the $\mathcal{X}_{SID}$ is :

$$\begin{aligned}
\mathbb{E}_{\tilde{\mathcal{E}}}[\frac{1}{n_{SID}} \sum_{x \in \mathcal{X}_{SID}} \frac{|\tilde{\mathcal{E}}|\ |\mathbf{X} = x}{x}] &= \frac{1}{n_{SID}} \sum_{x \in \mathcal{X}_{SID}} \frac{\mathbb{E}_{\tilde{\mathcal{E}}}[|\tilde{\mathcal{E}}|\ |\mathbf{X} = x]}{x} \\
&= \frac{1}{n_{SID}} \sum_{x \in \mathcal{X}_{SID}} \frac{\tilde{m}(x;\, \tilde{r}, \omega_{\mathcal{X}_{SID}})}{x} \\
&= \frac{\tilde{r}}{n_{SID}} \sum_{x \in \mathcal{X}_{SID}} \omega_{\mathcal{X}_{SID}}(x) \\
&= \tilde{r}
\end{aligned}$$

This is true with any weight function for which $\frac{1}{n_{SID}} \sum_{x \in \mathcal{X}_{SID}} \omega_{\mathcal{X}_{SID}}(x) = 1$. We now proceed to describe the construction of a sensible weight function.

3.3 Weight function for $\mathcal{X}_{SID} = \mathcal{X}$

We recall the plausibility criteria: we want our simulations of errors to be as close as possible to the population distribution. In particular, suppose that we want to do a simulation with a target MARE that happens to be the same as the MARE for the original data ($\mathcal{X}$ and $\mathcal{Y}$) and further suppose that we want to simulate using values from the entire data set (i.e., $\mathcal{X}_{SID} = \mathcal{X}$). Then we expect the simulated conditional distributions to be equal to the estimated conditional distributions. In other words,

$$\mathcal{X}_{SID} = \mathcal{X}, \quad \tilde{r} = r_{\hat{m}} \implies \forall x \in \mathcal{X}, \tilde{l}(x) = \hat{l}(x) \text{ and } \tilde{s}(x) = \hat{s}(x)$$

Solving the linear program (1) defined in subsection 3.1, leads to $\tilde{l}(x) = \hat{l}(x)$ and $\tilde{s}(x) = \hat{s}(x), \forall x \in \mathcal{X}$.

If we define the following $\hat{\omega}_{\mathcal{X}}$ function,

$$\forall x \in \mathcal{X},\ \hat{\omega}_{\mathcal{X}}(x) := \frac{\hat{m}(x)}{x r_{\hat{m}}} = \frac{\int_{\varepsilon=-\infty}^{\infty} |\varepsilon| \hat{f}_{\mathcal{E}|\mathbf{X}=x}(\varepsilon) d\varepsilon}{x r_{\hat{m}}}$$

First, we can verify that we have $\frac{1}{n} \sum_{x \in \mathcal{X}} \hat{\omega}_{\mathcal{X}}(x) = 1$. It is thus a weight function.

The choice of this weight function is natural when $\mathcal{X}_{SID} = \mathcal{X}$ because it is the ratio of the expected relative error simulated at x over the mean relative error when the errors are distributed according to the estimated joint distribution. However, choosing it when

$\mathcal{X}_{SID} \neq \mathcal{X}$ would satisfy our requirement for plausibility but it would prevent us from hitting the requested MARE.

Figure 3 illustrates that for the full CAISO wind dataset, the weight function presents a hyperbolic shape. The low values account for the biggest part of the MAPE.

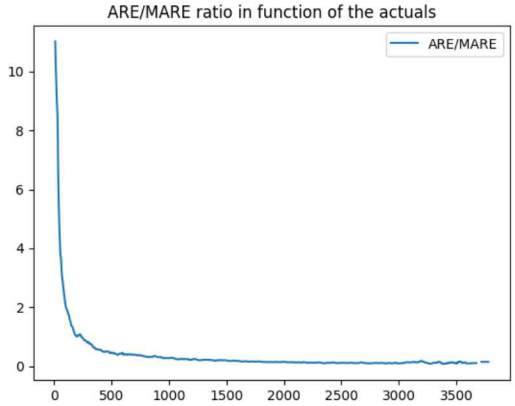


Figure 3: $\hat{\omega}_{\mathcal{X}}(x) = \frac{\hat{m}(x)}{x\hat{r}}$ ratio for the CAISO wind dataset.

3.4 Weight function for $\mathcal{X}_{SID}$ and an arbitrary $\tilde{r}$

Let us define the following Real that we call the plausibility score :

$$P_{\mathcal{X}_{SID}} = \frac{1}{n_{SID}} \sum_{x \in \mathcal{X}_{SID}} \hat{\omega}_{\mathcal{X}}(x)$$

When $\mathcal{X}_{SID} \neq \mathcal{X}$, the distribution of the SID is different from the distribution of the input dataset. Thus we do not necessarily have $P_{\mathcal{X}_{SID}} = 1$. A goal of our method is to meet the requested MARE, at least in expectation, no matter the $\mathcal{X}_{SID}$. If $P_{\mathcal{X}_{SID}}$ is greater than 1, it means that the distribution of $\mathcal{X}_{SID}$ has more data in the range where the weight function takes high values. This means that if use $\hat{\omega}_{datasetX}$, we are going to simulate too many errors with high values. While it has some physical sense, we are nonetheless going to simulate a greater MAPE than expected. Symmetrically, if $P_{\mathcal{X}_{SID}}$ is smaller than one, we are going to retrieve a lower MAPE than expected. This is illustrated in Figure 4), the density for the $\mathcal{X}_{SID}$ between December 2013 and March 2014 indicates more values at lower power than for the entire dataset, $\mathcal{X}$. If we simply used $\hat{\omega}_{\mathcal{X}}(x)$, for $x \in \mathcal{X}_{SID}$, then meeting the target AREs for each x would result in a MARE much greater than specified. In other words, since the ARE/MARE ratio is very high for the low power input, and since these inputs are over represented under the distribution of the December 2013 - March 2014 SID, we are going to simulate too many errors with a high target of mean absolute error. To meet the target MARE, a re-scaled weight function must thus be computed.

Let us define the following SID weight function :

$$\forall x \in \mathcal{X}_{SID},\ \tilde{\omega}_{\mathcal{X}_{SID}}(x) := \frac{\hat{\omega}_{\mathcal{X}}}{P_{\mathcal{X}_{SID}}}$$

With the re-scaled factor, we have $\frac{1}{n_{SID}} \sum_{x \in \mathcal{X}_{SID}} \tilde{\omega}_{\mathcal{X}_{SID}}(x) = 1$ so $\tilde{\omega}_{\mathcal{X}_{SID}} \in \Omega_{\mathcal{X}_{SID}}$.

Finally, for a given feasible $\tilde{r} \in \mathbb{R}_+$, we compute a $\tilde{\omega}_{\mathcal{X}_{SID}}$ which allocates the absolute errors across $\mathcal{X}_{SID}$ based on the allocation from $\mathcal{X}$. With these two parameters we can compute $\tilde{m}(x;\ \tilde{r}, \tilde{\omega}_{\mathcal{X}_{SID}}),\ x \in \mathcal{X}_{SID}$. According to Section 3.2.4, defining $\tilde{\mathcal{E}}$ from this target function, will get us $\mathbb{E}_{\tilde{\mathcal{E}}}[\frac{1}{n_{SID}} \sum_{x \in \mathcal{X}_{SID}} \frac{|\tilde{\mathcal{E}}|\ |\mathbf{X}=x}{x}] = \tilde{r}$.

We can also get the feasibility region for the target mare. For a given $\tilde{r}$ to be a feasible target mare, it must satisfy $\forall x \in \mathcal{X}, \tilde{m}(x, \tilde{r}, \tilde{\omega}_{\mathcal{X}_{SID}}) \leq m_{max}(x)$. Thus the feasibility region is :

$$\tilde{\mathcal{R}}_{\mathcal{X}_{SID}} = P_{\mathcal{X}_{SID}} \min \left(\frac{m_{max}(s)}{s\hat{\omega}_{\mathcal{X}}(s)}, s \in \mathcal{X}_{SID} \right)$$

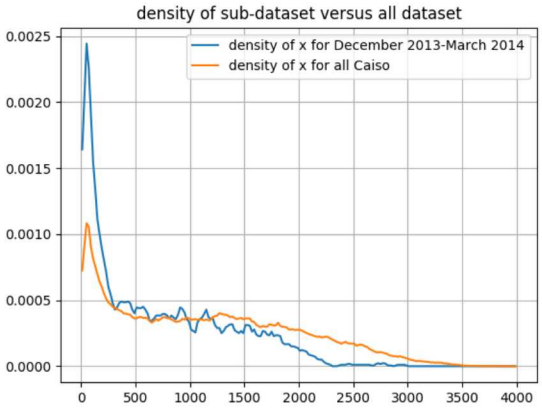


Figure 4: Comparison test density versus all dataset density

3.5 Simulating Without Auto-correlation

This is now straightforward. First, to obtain a simulation of errors, we are simulating a vector identically and independently distributed uniformly on $[0, 1]$, $(\tilde{\mathbf{U}}_t)_{t \leq n_{SID}}$. Then

$$\tilde{\mathcal{E}}_t = \tilde{F}^{-1}_{\mathcal{E}|\mathbf{X}=x_t}(\tilde{\mathbf{U}}_t), \quad \forall t \leq n_{SID}.$$

Let,

$$\tilde{\mathbf{Y}}_t = x_t + \tilde{F}^{-1}_{\mathcal{E}|\mathbf{X}=x_t}(\tilde{\mathbf{U}}_t), \quad \forall t \leq n_{SID}$$

so

$$\mathbb{E}[MARE(x, \tilde{\mathbf{Y}})] = \tilde{r}.$$

While we are hitting the target mare, the entire auto-correlation of the errors simulated relies solely on the auto-correlation of the input. In the extreme case where the errors are not depending on the input i.e $\tilde{F}_{\mathcal{E}|\mathbf{X}=x_i} = \tilde{F}_{\mathcal{E}|\mathbf{X}=x_0},\ \forall i \leq n_{SID}$ - which is the case for the middle power range for the CAISO wind data - then our simulations would have a null auto-correlation function. Implementing a base process to replace $(\tilde{\mathbf{U}}_t)_{t \leq n_{SID}}$ will generate the needed auto-correlation to satisfy the second point of the plausibility criteria.

4 Inferring a Base Process

The idea is to simulate a Base Process $\tilde{\mathbf{U}}_t$ of marginal Uniform in [0,1] depending on the past p lags $\tilde{U}_{t-i},\ i \leq p$ and the past q lags of errors over the base process $\delta_{t-i},\ i \leq q$.

Then, as previously in section 3.5, we would simulate the errors via the transformation $\hat{F}^{-1}_{\mathcal{E}|\mathbf{X}=x_t}(\tilde{\mathbf{U}}_t)$.

We model $Z_t = \phi^{-1}(U_t) \in (-\infty, \infty)$ as a Gaussian Process and more specifically as an ARMA process. Heuristically, we will show that this method gets us a good auto-correlation function for the simulations.

Inspired by the ARTA fit method (see [1]). We denote the CDF for the standard normal distribution ϕ and the CDF of the conditional distribution $\mathcal{E}|\mathbf{X} = x_t$, which is a beta distribution fit using $\mathcal{X}$, $\hat{F}_{\mathcal{E}|\mathbf{X}=x_t}$. Let us define the following time-series $(\hat{Z}_t)_t$:

$$\forall t \leq n, \quad \hat{Z}_t = \phi^{-1}(\hat{F}_{\mathcal{E}|\mathbf{X}=x_t}(\varepsilon_t))$$

We use the notation $(\hat{Z}_t)_t$ for the base process time-series of the dataset. Its empirical distribution is close to a standard Gaussian. Indeed, in section 2.3 we are estimating the conditional distribution so that $\mathcal{E}_t|\mathbf{X} = x_t$ has distribution that is approximated by $\hat{f}_{\mathcal{E}|\mathbf{X}=x_t}$, thus, $\hat{F}_{\mathcal{E}|\mathbf{X}=x_t}(\mathcal{E}_t) \dot{\sim} \mathcal{U}[0,1]$ and $\hat{\mathbf{Z}}_t \dot{\sim} \mathcal{N}(0,1)$. We fit on this base process an ARMA process. The standard definition of an ARMA process of order p and q uses $(a_i)_{i\leq p}$ and $(b_i)_{i\leq q}$ as coefficients so we temporarily reuse those symbols in this section.

Definition 4.1. $\{\mathbf{Z}_t\}$ is a base process if

- $\{\mathbf{Z}_t\}$ follows an ARMA process of order p and q :

$$\mathbf{Z}_t = \sum_{h=1}^{p} a_h \mathbf{Z}_{t-h} + \sum_{h=1}^{q} b_h \delta_{t-h} + \delta_t$$

  Where $\{\delta_t\}$ are the iid Gaussian error of mean 0 and variance σ_δ^2.

- $Var[\mathbf{Z}_t] = 1$, $E[\mathbf{Z}_t] = 0$, so that for all t, $\quad \mathbf{Z}_t \sim N(0,1)$.

We run a grid search over multiple (p, q) and we select the ARMA model to minimize the BIC criterion. The $(a_i)_{i\leq p}$ and $(b_i)_{i\leq q}$ found during the process define a function that enables us to generate base processes which will create the auto-correlation that we are looking for; however, fitting an ARMA imposes no constraint on the variance of noise σ_δ. So we are free to specify σ_δ so that we get $Var[\tilde{\mathbf{Z}}_t] = 1$.

Then, we can simulate directly the error by

$$\tilde{\mathcal{E}}_t = \tilde{F}^{-1}_{\mathcal{E}|\mathbf{X}=x_t}(\phi(\tilde{\mathbf{Z}}_t)), \quad \forall t \leq n_{SID}$$

and also get the result for the expected MARE established in section 3.5.

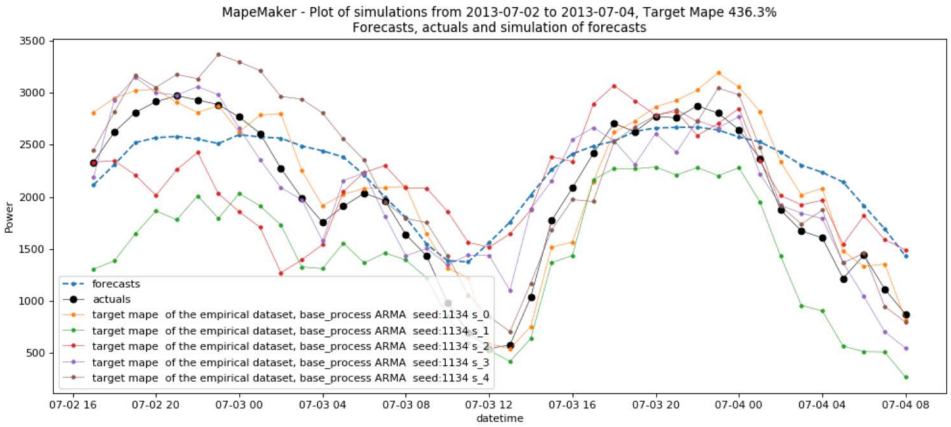
5 Enforcing Curvature

Let $(y_i)_i \in \mathbb{R}^n$ denote an simulation output time series. We define *curvature* at a point i in $(y_i)_i$ as

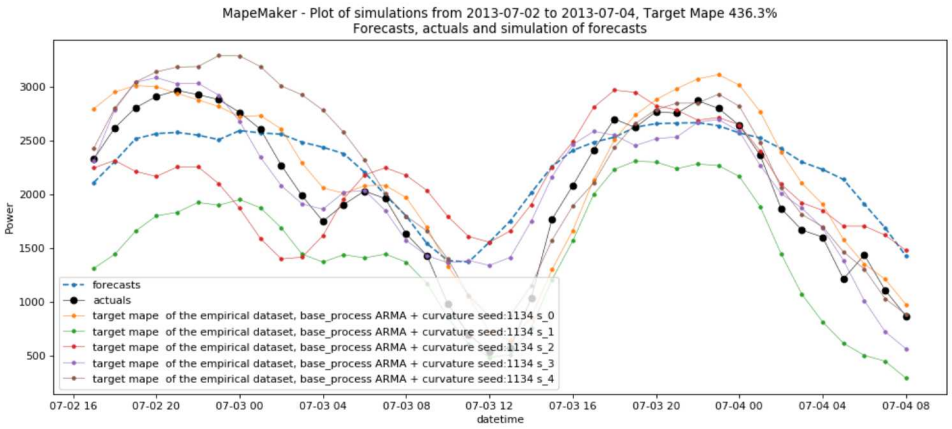
$$s_i = y_{i+2} - 2y_{i+1} + y_i \quad \forall i < n-2,$$

i.e., a second difference.

Methods described in Section 4 successfully model temporal correlation between the errors while satisfying a target MARE. However, some scenarios might not "look right" because of their lack of smooth curvature. This is especially unsatisfying in the case



(a) Illustration of forecast scenarios without curvature adjustment.



(b) Illustration of the same forecast scenarios with curvature adjustment

of forecasted renewables power production, which are much less sharp and erratic when compared to actual quantities.

We now concretely illustrate this issue with analysis of the CAISO wind power production data introduced previously. In Figure 5(a), we show baseline scenarios resulting from our proposed methods, i.e., without adjustment for curvature. As is clearly observed, the simulated forecasts in this case closely mirror the actuals – and not the one "true" forecast. In contrast, we show in Figure 5(b) a closely related set simulated forecasts – obtained by the procedure we now describe – that instead exhibit significantly more smooth and realistic curvature. Ultimately, the need for such adjustment depends entirely on the application.

In order to adjust the curvature of a forecast while still acheiving a target MARE, one approach is to *a posteriori* adjust a time series that already satisfies a target MARE such that specific curvature characteristics are imposed. We now formalize this general approach.

We introduce a minimization problem in which we penalize deviations from both a target second difference and the simulated forecast error. Per earlier analysis, we can simulate $(\tilde{\varepsilon}_i)_i$ using an ARMA base process. Then, define $d \in \mathbb{R}_+$, and let W_s and W_ε denote user inputs in $\mathbb{R}_+$.

We then let $(y_i)_i$ denote the solution of the following mathematical program:

$$\begin{aligned} \min_{y} \quad & \sum_{i=3}^{n} W_{s2}\Big(|y_i - 2y_{i-1} + y_{i-2}| - d\Big)^2 + W_\varepsilon\Big(y_i - x_i - \tilde{\varepsilon}_i\Big)^2 \\ \text{s.t.} \quad & y \in [0, cap]^n \end{aligned} \tag{3}$$

For practical computation, we now transform this mathematical program so that the objective function is quadratic and constraints are linear – such that widely available mathematical programming solvers can be leveraged. The transformation yields the following equivalent mixed-integer linear program (MILP), with $3n$ additional variables, n equality constraints, and $3n$ inequality constraints ($6n$ if we consider that the three real vectors are negatively bounded by 0):

$$\begin{aligned} \min_{y,\, \lambda^+,\, \lambda^-,\, b} \quad & \sum_{i=2}^{n} W_s\Big(\lambda_i^+ + \lambda_i^- - d\Big)^2 + W_\varepsilon\Big(y_i - x_i - \varepsilon_i\Big)^2 \\ \text{s.t.} \quad & y \in \mathbb{R}_+^n,\ \lambda^+ \in \mathbb{R}_+^n,\ \lambda^- \in \mathbb{R}_+^n,\ b \in \{0,1\}^n \\ & y_i \le cap \\ & \lambda_i^+ - \lambda_i^- = y_i - 2y_{i-1} + y_{i-2}, \quad \forall i \le n \\ & \lambda_i^+ \le b_i d_{max} \\ & \lambda_i^- \le (1 - b_i) d_{max} \end{aligned} \tag{4}$$

where d_{max} denotes a large constant; a safe value is $4cap$.

To verify equivalence of the two mathematical programs, we note that if $y_i - 2y_{i-1} + y_{i-2} \ge 0$, then because $b_i \in \{0, 1\}$, λ_i^- is equal to 0 with the two last equations. Then $\lambda_i^+ = y_i - 2y_{i-1} + y_{i-2}$ and $\lambda_i^+ + \lambda_i^- = y_i - 2y_{i-1} + y_{i-2}$. We use the same reasoning when $y_i - 2y_{i-1} + y_{i-2} < 0$.

Numerous open source and commercial solvers are available for such a mathematical program. However, solution time does generally increase with n. In many applications the restrictions on curvature are motivated by aesthetic or heuristic considerations. Thus, it can be reasonable to specify a "loose" optimality gap to avoid excessive computation time.

6 Putting it all Together

In this section we summarize the process to deliver a simulation with correct targets.

6.1 Procedures for estimation

First, as shown in Algorithm 1, we preprocess the data and estimate the conditional distributions using the methods explained in section 2.3. This results in a set of beta distribution parameters for each input from the whole dataset called $\hat{\mathcal{S}}$. To estimate the parameters we recall that the user should specify a data fraction (e.g., 0.05), for the sampling. (The software provides an option to produce a curve for the scores described in Section 2.4.)

Algorithm 1 Estimating the beta distributions

Input: x, y, a ▷ Input time-series and percent of data
Output: $\hat{\mathcal{S}}_x$

procedure COMPUTING_ESTIMATION_PARAMETERS(x, y, a)
  $\mathcal{X} \leftarrow sort(x)$
  for $x \in \mathcal{X}$ **do** ▷ Applying the methodology explained in section 2.3
    Compute the interval of estimation I_x^a and sample E_x^a .
    $\bar{x}(x,a) \leftarrow \bar{I_x^a}$
    $\hat{l}(\bar{x}(x,a)), \hat{s}(\bar{x}(x,a)) \leftarrow Bounds(E_x^a, x)$ ▷ See section 2.3.1
    $\hat{\alpha}(\bar{x}(x,a)), \hat{\beta}(\bar{x}(x,a)) \leftarrow Moments(mean(E_x^a), std(E_x^a))$ ▷ See section 2.3.2
  for $x \in \mathcal{X}$ **do** ▷ Take the closest computed point of estimation
    $x' \leftarrow argmin|\bar{x}(x',a) - x|$
    $\hat{\mathcal{S}}_x \leftarrow (\hat{\alpha}, \hat{\beta}, \hat{l}, \hat{s})(\bar{x}(x',a))$
  return $\hat{\mathcal{S}}$

Next, as shown in Algorithm 2, we estimate the partitioning of the mean absolute percent errors according to the input and we encode this information in the weight function. An important feature of this procedure is the computation of $r_{\hat{m}}$ which is the expected mean absolute relative error from the conditional distributions (which may be close in value to, but is different from, $\hat{r}$.) This procedure is explained in section 3.3.

The next phase, shown in Algorithm 3, is estimation of the underlying base_process that generates auto-correlation in the time-series of the errors. This is done by using the CDF B of the beta distribution whose parameters have been inferred in step 1. Then we operate a grid search over the p and q parameters to select the order of the model that minimize the BIC criterion. We save the coefficients. Recalling that we want the marginal of $Z \sim \mathcal{N}(0,1)$, we set the variance of the errors of the base process so that $Var[\mathbf{Z}_t] = 1$. This procedure is explained in section 4.

6.2 Procedures to deliver the target mare

First, as shown in Algorithm 4, given a target mare $\tilde{r}$, and a $\mathcal{X}_{SID}$ we verify that $\tilde{r}$ is feasible. If it is, we aim at targeting a mean absolute error for each conditional distribution with input in the $\mathcal{X}_{SID}$. For this we compute a target function using the estimated weight function (see section 3.4).

Algorithm 2 Estimating the weight function

Input: $\hat{\mathcal{S}}_x$, $\mathcal{X}$
Output: $\omega_{\mathcal{X}}$

procedure COMPUTING_ESTIMATED_WEIGHT_FUNCTION($\hat{\mathcal{S}}_x$, $\mathcal{X}$)
  $r_{\hat{m}} \leftarrow 0$
  for $x \in \mathcal{X}$ **do** ▷ Applying the methodology explained in section 3.3
    $\hat{m}(x) \leftarrow \int_{\varepsilon=-\infty}^{\infty} |\varepsilon| beta(\varepsilon; \mathcal{S}_{\mathcal{X}}(x)) d\varepsilon$.
    $m_{max}(x) \leftarrow \max \nu(l, s, \hat{\alpha}(x), \hat{\beta}(x))$ ▷ See constraints on target function (2)
    $\omega_{\mathcal{X}}(x) \leftarrow \frac{\hat{m}(x)}{x}$
    $r_{\hat{m}} \leftarrow r_{\hat{m}} + \hat{m}(x)$
  $r_{\hat{m}} \leftarrow \frac{r_{\hat{m}}}{|\mathcal{X}|}$
  $\omega_{\mathcal{X}} \leftarrow \frac{\omega_{\mathcal{X}}}{r_{\hat{m}}}$
  return $\omega_{\mathcal{X}}$

Algorithm 3 Fitting the Base Process ARMA process

Input: x, $\mathcal{E}$, $\hat{\mathcal{S}}_x$
Output: $(a_i)_{i \leq p}$, $(b_i)_{i \leq q}$, σ_δ

procedure FIT_ARMA_PROCESS(x, $\mathcal{E}$, $\hat{\mathcal{S}}_x$)
  for $i \in [1, len(x)]$ **do** ▷ Estimating the base process see section 4
    $\hat{Z}_i \leftarrow \phi^{-1}(B(\varepsilon_i, \mathcal{S}_{\mathcal{X}}(x_i)))$
  $BIC \leftarrow +\infty$
  $p,\ q \leftarrow 0,\ 0$
  for $p', q' \in [0, 5]^2$ **do** ▷ Grid Searching
    $tempBIC \leftarrow BIC(ARMA(\hat{Z}, (p', 0, q'))$
    if $tempBIC < BIC$ **then**
      $BIC \leftarrow tempBIC$
      $p,\ q \leftarrow p',\ q'$
  $(a_i)_{i \leq p},\ (b_i)_{i \leq q} \leftarrow ARMA(\hat{Z}, (p, 0, q))$
  $\sigma_\delta \leftarrow \text{argmin}_\sigma(std(ARMA((a_i)_{i \leq p},\ (b_i)_{i \leq q}, \sigma) - 1)^2$
  return $(a_i)_{i \leq p},\ (b_i)_{i \leq p},\ \sigma_\delta$

Algorithm 4 Inferring a target function for the SID

Input: $\hat{\mathcal{S}}$, $\mathcal{X}_{SID}, \tilde{r}, \hat{\omega}_{\mathcal{X}}$
Output: $\tilde{m}$

procedure COMPUTING_SIMULATION_TARGET_FUNCTION($\hat{\mathcal{S}}$, $\mathcal{X}_{SID}, \tilde{r}, \hat{\omega}_{\mathcal{X}}$)
  $P_{\mathcal{X}_{SID}} \leftarrow 0$ ▷ Computing the Plausibility score
  for $s \in \mathcal{X}_{SID}$ **do**
    $P_{\mathcal{X}_{SID}} \leftarrow P_{\mathcal{X}_{SID}} + \frac{\hat{\omega}_{\mathcal{X}}(s)}{|\mathcal{X}_{SID}|}$.
  $\tilde{r}_{max} \leftarrow P_{\mathcal{X}_{SID}} \min\left(\frac{m_{max}(s)}{s\hat{\omega}_{\mathcal{X}}(s)}, s \in \mathcal{X}_{SID}\right)$
  if $\tilde{r} > \tilde{r}_{max}$ **then**
    Report Error
  $\tilde{\omega}_{\mathcal{X}_{SID}} \leftarrow \frac{\hat{\omega}_{\mathcal{X}}}{P_{\mathcal{X}_{SID}}}$
  for $s \in \mathcal{X}_{SID}$ **do** ▷ Applying the function as explained in section 3.4
    $\tilde{m}(s) \leftarrow \tilde{r} s \tilde{\omega}_{\mathcal{X}_{SID}}(s)$
  return $\tilde{m}$

Second, as shown in Algorithm 5, according to a target function $\tilde{m}$, we assign adjusted parameters for each conditional distribution whose input is in the $\mathcal{X}_{SID}$. We move the location parameters from the estimated ones while keeping the shape parameters. See section 3.2.

Algorithm 5 Inferring the simulation beta distributions

Input: $\tilde{m}$, $\mathcal{S}$, $\mathcal{X}_{SID}$
Output: $\tilde{\mathcal{S}}_{\tilde{m}}$

procedure ADJUSTING_SIMULATION_PARAMETERS($\tilde{m}$, $\mathcal{S}$, $\mathcal{X}_{SID}$)
  for $\delta \in \mathcal{X}_{SID}$ **do** ▷ Applying the methodology explained in section
    $x \leftarrow closest(\delta, \mathcal{X})$ ▷ not necessarily $\mathcal{X}_{SID} \subset \mathcal{X}$
    $\tilde{\alpha}(\delta), \tilde{\beta}(\delta) \leftarrow \hat{\alpha}(x), \hat{\beta}(x)$
    $\tilde{l}(\delta), \tilde{s}(\delta) \leftarrow Program_1(\tilde{\alpha}(\delta), \tilde{\beta}(\delta), \delta, \tilde{m}(\delta))$ ▷ See equation 1
    $\tilde{\mathcal{S}}_{\delta,\tilde{m}} = (\tilde{\alpha}(\delta), \tilde{\beta}(\delta), \tilde{l}(\delta), \tilde{s}(\delta))$
  return $\tilde{\mathcal{S}}_{\tilde{m}}$

6.3 Procedure to simulate the output

Using methods summarized in Algorithm 6, we simulate a base process sample of length $|\mathcal{X}_{SID}|$ and use the simulated conditional distributions to obtain conditioned errors. We directly get the simulation by summing the errors and the input data. Finally, if the user asks for it, we optimize the curvature *a posteriori*, see section 5.

Algorithm 6 Simulating a sample of output

Input: $\tilde{m}$, $\mathcal{S}$, $\mathcal{X}_{SID}, (a_i)_{i\leq p}$, $(b_i)_{i\leq p}$, σ_δ, $\tilde{\mathcal{S}}_{\mathcal{X}_{SID}}$, which implies $\tilde{F}^{-1}_{\mathcal{E}|\mathbf{X}=x_t}$
Output: $(\tilde{y}_i)_{i\leq n_{SID}}$

procedure COMPUTING_ESTIMATION_PARAMETERS($\tilde{m}, \mathcal{S}, \mathcal{X}_{SID}, (a_i)_{i\leq p}, (b_i)_{i\leq p}, \sigma_\delta$)
  $(\tilde{z}_i)_{i\leq n_{SID}} \leftarrow createArmaSample((a_i)_{i\leq p}, (b_i)_{i\leq p}, \sigma_\delta, n_{SID})$
  for $i \in [1, n_{SID}]$ **do**
    $\tilde{\varepsilon}_i = \tilde{F}^{-1}_{\mathcal{E}|\mathbf{X}=x_t}(\phi(z_i))$
    $\tilde{y}_i = x_i + \tilde{\varepsilon}_i$
  if Curvature is $True$ **then**
    $(\tilde{y}_i)_{i\leq n_{SID}} \leftarrow Optimization_1(\tilde{\varepsilon}, d, x, cap)$ ▷ See Program 4
  return $(\tilde{y}_i)_{i\leq n_{SID}}$

7 Evaluation

We used computed scores to evaluate our simulations based on the similarity with the empirical data and with the satisfaction of the target. We want to assess the quality of the convergence of the metrics with respect to : the number of scenarios simulated (let us denote it M), the length of the input array (let us denote it n_t), the type of simulation.

We study three types of simulations:

- A) IID base process, ϕ_1

- B) ARMA base process, ϕ_2
- C) ARMA base process and curvature optimization, ϕ_3

7.1 Target MAPE Score

The score function for achieving the target MAPE is

$$S_{mare}(M, n_t, k) = \sqrt{\sum_{i=1}^{M} (\tilde{r} * 100\% - MAPE((x_i)_{i \leq n_t}, \phi_k((x_i)_{i \leq n_t})))^2}.$$

7.2 Validation of the base process

Let p be the maximum lag of auto-correlation we wish to assess.

Let us define the functional

$$\hat{\rho}((\varepsilon_i)_{i \leq n}, j) = \frac{1}{(n-j)\sigma^2} \sum_{i=0}^{n-j} \varepsilon_{i+j} \varepsilon_i$$

If $\bar{\varepsilon} = 0$, $\hat{\rho}((\varepsilon_i)_{i \leq n}, j)$ is the estimation of the auto-correlation of the errors of the input dataset at lag j. $\hat{\rho}((\phi_k(x_i) - x_i)_{i \leq n_t}, j)$ is the estimation of the auto-correlation at lag j of the errors simulated by the MAPE_maker of type k with an SID starting at the beginning of the dataset and of length n_t.

$$S_{auto\_correlation}(M, n_t, k, p) = \sqrt{\sum_{i=1}^{M} \sum_{j=1}^{p} (\hat{\rho}((\varepsilon_i)_{i \leq n}, j) - \hat{\rho}((\phi_k(x_i) - x_i)_{i \leq n_t}, j))^2}$$

7.3 Validation of the curvature

Let us define the functional

$$D((y_i)_{i \leq n}) = \frac{1}{n-2} \sum_{i=0}^{n-2} y_i - 2y_{i-1} + y_{i-2}$$

$$S_{second\_difference}(M, n_t, k) = \sqrt{\sum_{i=1}^{M} (D((y_i)_{i \leq n}) - D(\phi_k(x_i)_{i \leq n_t}))^2}$$

7.4 Score function

The score function is the sum of those three targets weighted :

$$\begin{aligned} S(M, n_t, k, p;\ w_m, w_{ac}, w_{sd}) = {} & w_m * S_{mare}(M, n_t, k) + \\ & w_{ac} * S_{auto\_correlation}(M, n_t, k, p) + \\ & w_{sd} * S_{second\_difference}(M, n_t, k) \end{aligned}$$

7.5 Behavior of the scores as M grows

To illustrate the behavior of the simulated scenarios as the number of scenarios created, M, grows we conducted experiments using the CAISO wind dataset and created scenarios for three days. Figure 5 shows that for this example, the achieved MARE is close to the target MARE as soon as there are about 4 scenarios. The scenarios that use curvature correction result in a value that is closest to the target, which makes sense because Program 4 corrects for the MARE after the scenarios are created. However, the other scenarios are reasonably close. Figure 6 demonstrates that for processes that are not iid, the autocorrelation score is quit good almost regardless of the number of scenarios. Figure /reffig:curveconverge shows that the curvature (second differences) score does not depend on the number of scenarios generated and that the methods are ordered as expected.

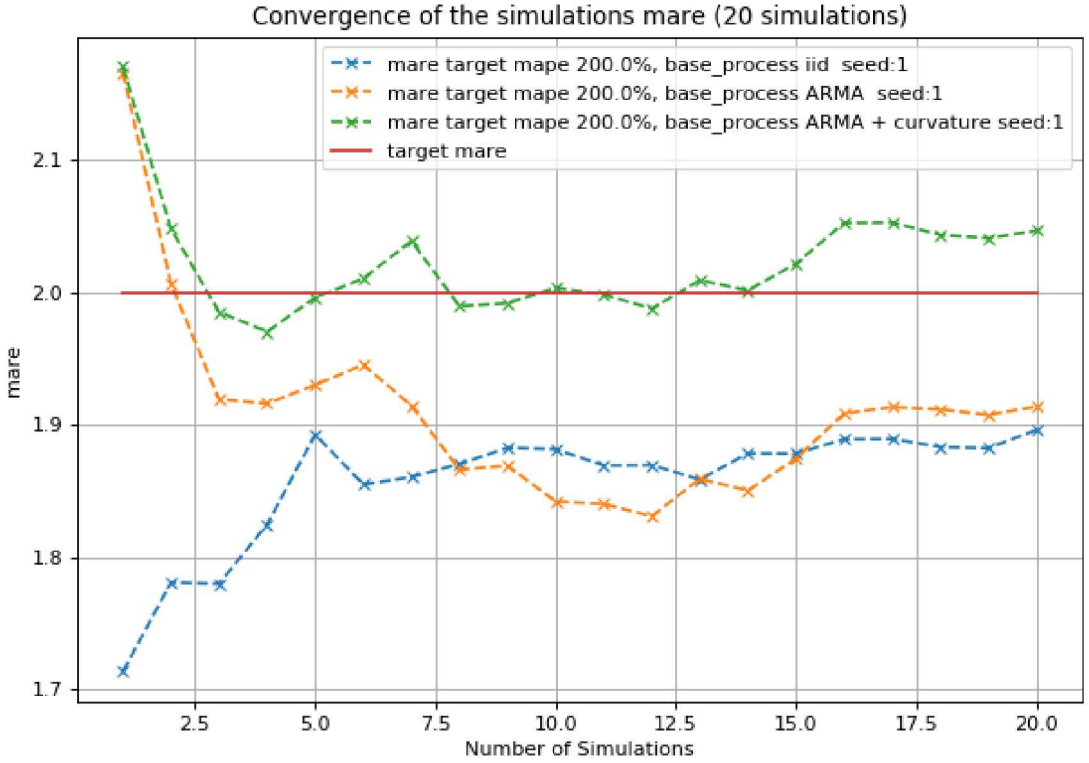


Figure 5: MARE score as a function of the number of scenarios created by simulation.

8 Conclusions

We have described methods for creating scenarios that make use of a history of forecast errors. The corresponding software is available for download and use. Although we used wind data from CAISO in our illustrations, the method can be used for any situation where there is a history of forecasts and actuals. In particular, the software has been used to create scenarios for load, solar, and wind for the rts-gmlc data `https://github.com/GridMod/RTS-GMLC`.

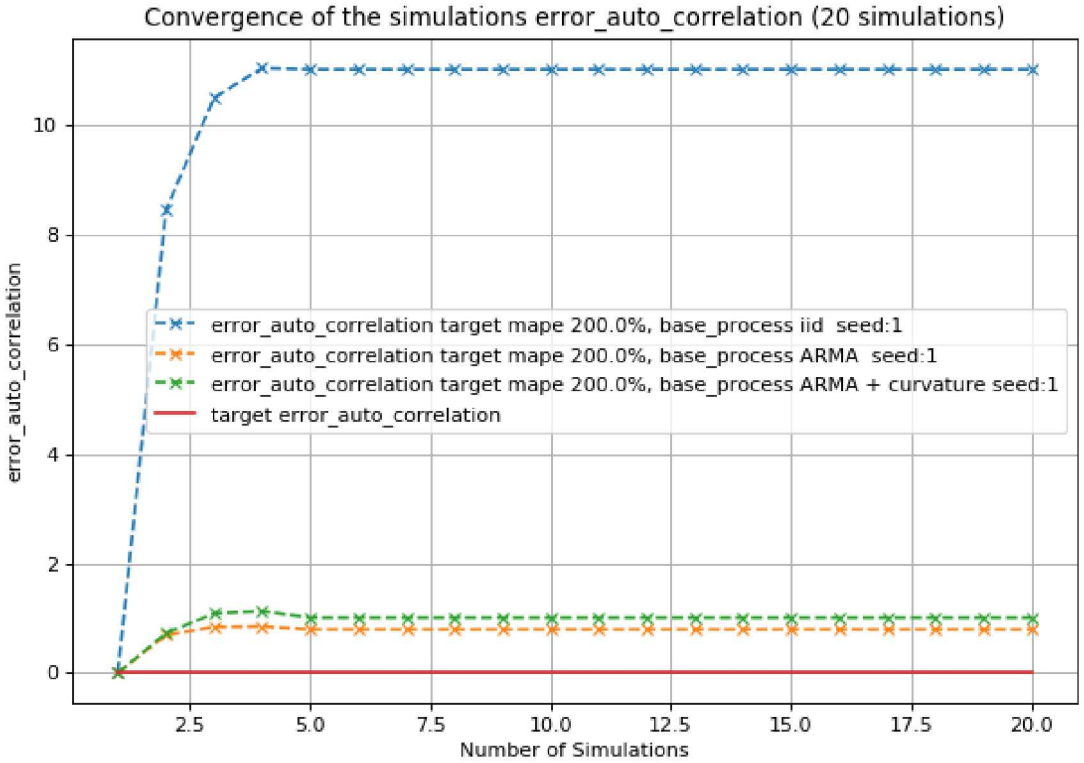


Figure 6: Sum of the absolute difference of the auto-correlation score as a function of the number of scenarios created by simulation.

The use of solar requires pre- and post-processing of the input data to work well. Instead of power values, the forecasts and actuals should be presented as fractions of capacity and with the value of *cap* set to one during the day and zero at night. This is because solar power is always zero at night and because the concept of "low power" changes during the day.

Future research includes consideration of error measures other than the MAPE. On the purely software front, we are working to parallelize computations. The software and the methods described here are intended to be an addition to the kit of tools available for dealing with uncertainty in power generation planning and operations.

Acknowledgement

Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract `DE-NA0003525`. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

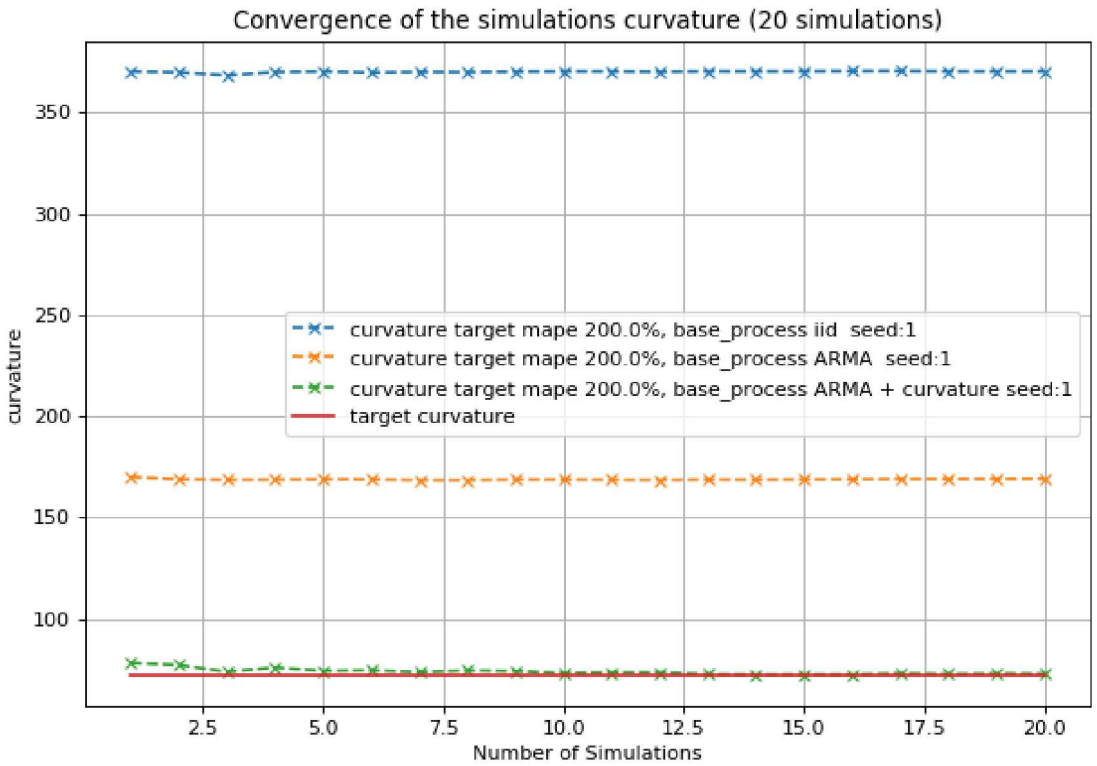


Figure 7: Second difference score as a function of the number of scenarios created by simulation.

References

[1] Bahar Biller and Barry L. Nelson. Fitting time-series input processes for simulation. *Operations Research*, 53:549–559, 2005.

[2] Y. Dvorkin, Y. Wang, H. Pandzic, and D. Kirschen. Comparison of scenario reduction techniques for the stochastic unit commitment. In *2014 IEEE PES General Meeting — Conference Exposition*, pages 1–5, July 2014.

[3] P. Pinson, G. Kariniotakis, H. A. Nielsen, T. S. Nielsen, and H. Madsen. Properties of quantile and interval forecasts of wind generation and their evaluation. In *Proceedings of the European Wind Energy Conference & Exhibition*, Athens, 2006. http://www.ewea.org.

[4] P Pinson, H Madsen, H A Nielsen, G Papaefthymiou, and B Klöckl. From probabilistic forecasts to statistical scenarios of short-term wind power production. *Wind Energy*, 12:51–62, 2008.

[5] Pierre Pinson and Robin Girard. Evaluating the quality of scenarios of short-term wind power generation. *Applied Energy*, 96:12–20, 2012.

[6] Didem Sari, Youngrok Lee, Sarah Ryan, and David Woodruff. Statistical metrics for assessing the quality of wind power scenarios for stochastic unit commitment. *Wind Energy*, 19(5):873–893, 2016.

[7] Chris Tofallis. A better measure of relative prediction accuracy for model selection and model estimation. *Journal of the Operational Research Society*, 66(8):1352–1362, 2015.

[8] David L. Woodruff, Julio Deride, Andrea Staid, Jean-Paul Watson, Gerrit Slevogt, and César Silva-Monroy. Constructing probabilistic scenarios for wide-area solar power generation. *Solar Energy*, 160:153 – 167, 2018.