Long Term Time Series Generation Conditioned on Sequential Short-Term Forecasts

Kyle Perline

This code implements the Long Term Generation algorithm. This algorithm was developed in Kyle Perline's Ph.D. thesis at Cornell University, and this work is currently under submission to IEEE Transactions on Sustainable Energy.

The Long Term Generation algorithm generates long term time series conditioned on sequential short-term historical forecasts. The application is to generating wind power scenarios conditioned on sequential short-term historical wind power forecasts.

1 Technical Definitions and Objective

Let the time index be t. Let R_t be a (univariate) response random variable. Let $F_t = (F_t^1, ..., F_t^\Delta)$ be a Δ-dimensional random vector of forecasts, where F_t^i predicts the response variable R_{t+i}, for each $i = 1, ..., \Delta$. (The predictor variables F_t can be more general than the described forecasts; e.g. they can be probabilistic forecasts instead of point forecasts, or they do not even need to be forecasts at all.) Over time steps $T_1, ..., T_2$ there is some joint probability density function P of the R_t and F_t, denoted

$$P(F_{T_1}, R_{T_1}, F_{T_1+1}, R_{T_1+1}, ..., F_{T_2}, R_{T_2}).$$

Let r_t and f_t be historical samples of the random variables R_t and F_t, respectively.

Suppose that historical forecasts $f_{T_1}, ..., f_{T_2}$ are obtained. Then the conditional distribution of the response variables conditioned on the forecasts is

$$P(R_{T_1}, ..., R_{T_2} | f_{T_1}, ..., f_{T_2}).$$

The Long Term Generation algorithm creates an estimate $\hat{P}$ of this conditional distribution, i.e.

$$\hat{P}(R_{T_1}, ..., R_{T_2} | f_{T_1}, ..., f_{T_2}) =_d P(R_{T_1}, ..., R_{T_2} | f_{T_1}, ..., f_{T_2}).$$

This Long Term Generation algorithm has three main steps:

1. For each time step $t = T_1, ..., T_2$ estimate the **marginal distribution** $P(R_t | f_{T_1}, ..., f_{T_2})$.

   This is accomplished by:

   (a) Create a predictor random variable $X_{t,u} = h_{t,u}(F_{T_1+u}, ..., F_{T_2+u})$; let $x_{t,u}$ be the historical sample.

   (b) For some positive integer, draw historical samples

   $$D_t = \{(x_{t,u}, r_t)\}_{-N \leq u \leq N}.$$

   This definition assumes that P is *slowly time varying.*

   (c) Use a numerical method to construct the estimate $\hat{P}(R_t | X_{t,0})$ based on D_t. The two numerical methods that have been implemented are *Kernel Density Estimation* and *Quantile Regression.*

   For example, if $X_{t,u} = F_{t-1+u}^1$, then the predictor variable is the previous time step's one time step ahead forecast. Then in step (b) the data set at time t is

   $$D_t = \{(f_{t-1+u}^1, r_{t+u})\}_{-N \leq u \leq N}.$$

   In step (c) the slowly time varying assumption means that each data point in D_t is drawn from the same distribution $P(R_t | F_{t-1}^1)$. We therefore construct the estimate $\hat{P}(R_t | F_{t-1}^1)$.

2. Construct the **joint distribution** $\hat{P}(R_{T_1}, ..., R_{T_2} | X_{T_1,0}, ..., X_{T_2,0})$ based on the marginal distributions. A Gaussian copula approach is used, where each of the marginal distributions are transformed into a standard normal distribution. The joint distribution is then uniquely specified by the covariance matrix. There are multiple methods for estimating or constructing this covariance matrix.

3. Draw scenarios from $\hat{P}(R_{T_1}, ..., R_{T_2} | X_{T_1,0}, ..., X_{T_2,0})$.

2 Long Term Generation code

There are three main sections of code.

1. Marginal Distributions: Both Kernel Density Estimation (KDE) and Quantile Regression (QR) have been implemented. Run `KDEdemo` for an example of KDE usage.
2. Joint Distribution: The Gaussian copula method has been implemented, with various approaches for estimating the covariance matrix.
3. Getting Data: Various methods were implemented for automatically obtaining the data set D_t in step 1b.

Wind scenarios can be generated using the Long Term Generation algorithm by calling `RUN_GEN_SCENARIOS.LTG()`.