Scenario generation for nongaussian time series via Quantile Regression

Marcelo Ruas e Alexandre Street

November 16, 2017

Motivation

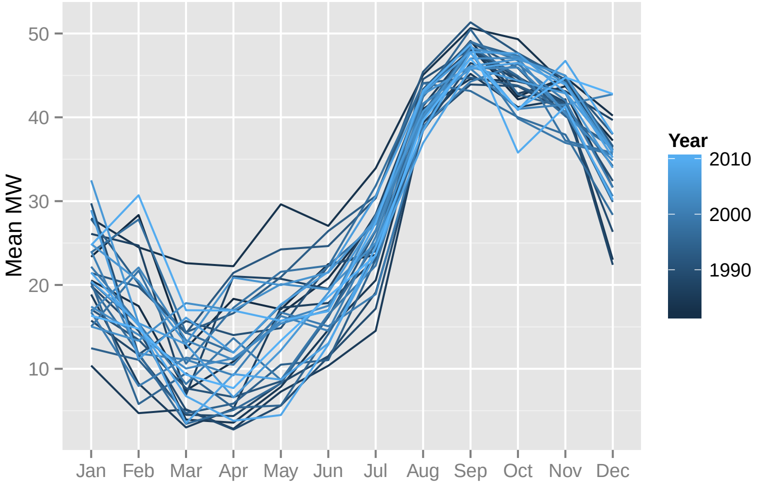
- Renewable energy scenarios are important in many fields in Power Systems:
  1. Energy trading;
  2. unit commitment;
  3. grid expansion planning;
  4. investment decisions
- In stochastic optimization problems, a set of scenarios is a needed input.
- Robust optimization requires bounds for probable values.

Change in paradigm: from predicting the conditional mean to predicting the conditional distribution

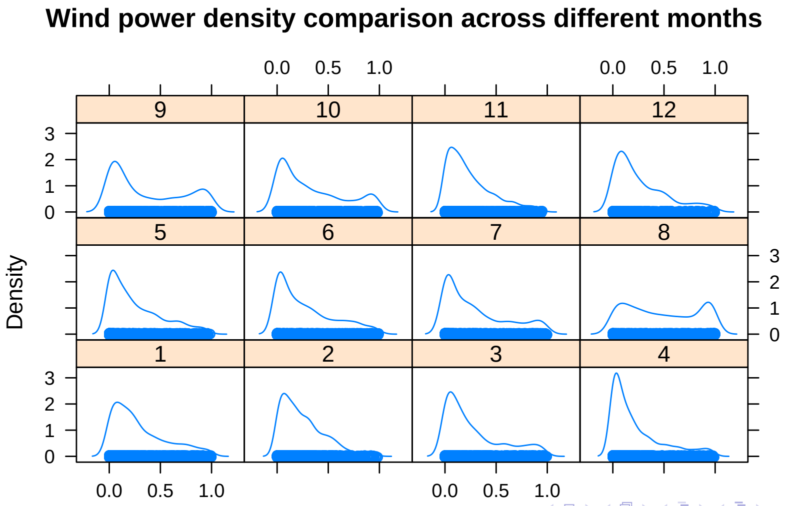
Probability Forecasting Approaches

- *Parametric Models*
  - Assume a distributional shape
  - Low computational costs
  - Faster convergence
  - *Examples: Arima-GARCH, GAS*
- *Nonparametric Models*
  - Don't require a distribution to be specified
  - High computational cost
  - Needs more data to produce a good approximation
  - *Examples: Quantile Regression (Koenker and Bassett Jr (1978)), Kernel Density Estimation (Gallego-Castillo et al. (2016)), Artificial Intelligence (Wan et al. (2017))*

Wind Power Time Series - Icaraizinho monthly data



Wind Power Time Series - Kaggle forecasting competition hourly data



The nongaussianity of Wind Power

- Renewables, such as wind and solar power have reportedly nongaussian behaviour
- Convenience of using a nonparametric approach, which doesn't rely on assuming a distribution
- Quantile regression is the chosen technique available to model this time series dynamics, by estimating a thin grid of α-quantiles at once and forming a data-driven conditional distribution

Objectives

- A nonparametric methodology to model the conditional distribution of renewables time series to produce scenarios.
- We propose a methodology that selects the global optimal solution with parsimony both on the selection of covariates as on the quantiles. Regularization methods are based on two techniques: Best Subset Selection (MILP) and LASSO (Linear Programming)
- Regularization techniques applied to an ensemble of quantile functions to estimate the conditional distribution, solving the issue of non-crossing quantiles. On regularizing quantiles, we propose a smoothness on the coefficients values across the sequence of quantiles.

Definition of the Conditional Quantile

Let the conditional quantile function of Y for a given value x of the d-dimensional random variable X, i.e., $Q_{Y|X} : [0,1] \times \mathbb{R}^d \to \mathbb{R}$, can be defined as:

$$Q_{Y|X}(\alpha, x) = F^{-1}_{Y|X=x}(\alpha) = \inf\{y : F_{Y|X=x}(y) \geq \alpha\}.$$

Conditional Quantile from a sample

Let a dataset be composed from $\{y_t, x_t\}_{t \in T}$ and let ρ be the check function

$$\rho_\alpha(x) = \begin{cases} \alpha x & \text{if } x \geq 0 \\ (1-\alpha)x & \text{if } x < 0 \end{cases}, \tag{1}$$

The sample quantile function for a given probability α is then based on a finite number of observations and is the solution to minimizing the loss function $L(\cdot)$:

$$\hat{Q}_{Y|X}(\alpha, \cdot) \quad \in \quad \underset{q(\cdot) \in \mathcal{Q}}{\arg\min}\, L_\alpha(q) = \sum_{t \in T} \rho_\alpha(y_t - q(x_t)),$$

$$q(x_t) = \beta_0 + \beta^T x_t,$$

where $\mathcal{Q}$ is a space of functions. In this paper, we use $\mathcal{Q}$ as an **affine functions space**.

Conditional Quantile from a sample

- For a single quantile, this problem can be solved by the following Linear Programming problem:

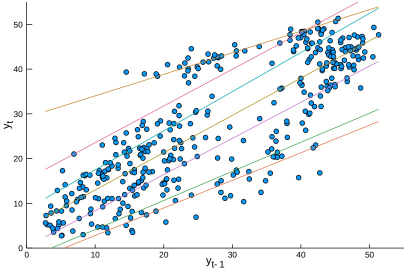
$$
\begin{array}{lll}
\min\limits_{\beta_0,\beta,\varepsilon_t^+,\varepsilon_t^-} & \sum_{t \in T} \left(\alpha \varepsilon_t^+ + (1-\alpha)\varepsilon_t^- \right) & \\
\text{s.t.} & \varepsilon_t^+ - \varepsilon_t^- = y_t - \beta_0 - \beta^T x_t, & \forall t \in T, \\
 & \varepsilon_t^+, \varepsilon_t^- \geq 0, & \forall t \in T.
\end{array}
$$

- The output are the coefficients β_0 and β (which is the same dimension as x_t), that describe the quantile function as an affine function.

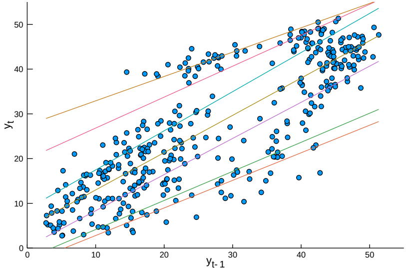
The non-crossing issue

- The following condition must always hold:

$$q_{\alpha}(x_t) \leq q_{\alpha'}(x_t), \text{ when } \alpha \leq \alpha'$$



(a) Each α-quantile estimated independently



(b) Estimation with non-crossing constraint

Figure: These graphs show how the addition of a constraint can contour the crossing quantile issue

Notation

Expression	Meaning
$Q_{Y\|X}(\alpha, x)$	The conditional quantile function
y_t	the time series we are modelling
x_t	explanatory variables of y_t in t
T	the set containing all observations indexes
J	the set containing all quantile indexes
$J_{(-1)}$	the set $J \backslash \{1\}$
α_j	a probability, might be indexed by j
A	the set of probabilities $\{\alpha_j \mid j \in J\}$
K	Maximum number of covariates on MILP regularization
λ	The Lasso penalization on the coefficients ℓ_1-norm
γ	The penalization on the coefficients second-derivative with respect of the quantiles

Conditional Quantile as a Linear Programming Problem

$$\min_{\beta_{0j},\beta_j,\varepsilon_{tj}^+,\varepsilon_{tj}^-} \sum_{j\in J}\sum_{t\in T}\left(\alpha_j\varepsilon_{tj}^+ + (1-\alpha_j)\varepsilon_{tj}^-\right)$$

$$\begin{aligned}
&\text{s.t.} && \\
&\varepsilon_{tj}^+ - \varepsilon_{tj}^- = y_t - \beta_{0j} - \beta_j^T x_t, && \forall t\in T, \forall j \in J,\\
&\varepsilon_{tj}^+, \varepsilon_{tj}^- \geq 0, && \forall t\in T, \forall j \in J,\\
&\beta_{0,j-1} + \beta_{j-1}^T x_t \leq \beta_{0j} + \beta_j^T x_t, && \forall t\in T, \forall j \in J_{(-1)},
\end{aligned}$$

- Coefficients β_{0j} and β_j refer to the j^{th} quantile
- We apply QR to estimate the conditional distribution $\hat{Q}_{Y_{t+h}|X_{t+h},Y_t,Y_{t-1},\dots}(\alpha,\cdot)$ for a k-step ahead forecast of time serie $\{y_t\}$, where X_{t+h} is a vector of exogenous variables at the time we want to forecast.

Best Subset selection via MILP

- Mixed Integer Linear Programming (MILP) models allow only K variables to be used for each α-quantile.
- Only K coefficients β_{pj} may have nonzero values, for each α-quantile.
- It is guaranteed by constraints on the optimization model.
- One model for each α-quantile

Best Subset selection via MILP

$$
\begin{aligned}
\min_{\beta_{0j},\beta_j,z_{pj},\varepsilon^+_{tj},\varepsilon^-_{tj}} \quad & \sum_{j \in J}\sum_{t \in T}\left(\alpha_j \varepsilon^+_{tj} + (1-\alpha_j)\varepsilon^-_{tj}\right) & \\
\text{s.t} \quad & \varepsilon^+_{tj} - \varepsilon^-_{tj} = y_t - \beta_{0j} - \beta_j^T x_t, & \forall t \in T, \forall j \in J, \\
& \varepsilon^+_{tj}, \varepsilon^-_{tj} \geq 0, & \forall t \in T, \forall j \in J, \\
& -Mz_{pj} \leq \beta_{pj} \leq Mz_{pj}, & \forall j \in J, \forall p \in P, \\
& \sum_{p \in P} z_{pj} \leq K, & \forall j \in J, \\
& z_{pj} \in \{0,1\}, & \forall j \in J, \forall p \in P, \\
& \beta_{0,j-1} + \beta_{j-1}^T x_t \leq \beta_{0j} + \beta_j^T x_t, & \forall t \in T, \forall j \in J_{(-1)},
\end{aligned}
$$

- z_{pj} is a binary variable which indicates when $\beta_{pj} > 0$.

Variable Selection via LASSO

- Regularization by including the coefficients ℓ_1-norm on the objective function.
- In this method, coefficients are shrunk towards zero by changing a continuous parameter λ, which penalizes the size of the ℓ_1-norm.
- When the value of λ gets bigger, fewer variables are selected to be used.
- The optimization problem for a single quantile is presented below:

$$\min_{\beta_0,\beta} \sum_{t \in T} \rho_\alpha(y_t - (\beta_0 + \beta^T x_t)) + \lambda \|\beta\|_1,$$

Variable Selection via LASSO

- At first, we select variables using LASSO

$$\begin{aligned}
&\underset{\beta_0,\beta,\varepsilon^+_{tj},\varepsilon^-_{tj}}{\text{arg min}} \sum_{j\in J}\sum_{t\in T}\left(\alpha_j\varepsilon^+_{tj}+(1-\alpha_j)\varepsilon^-_{tj}\right)+\lambda\sum_{p\in P}\xi_{pj}\\
&\text{subject to}\\
&\varepsilon^+_{tj}-\varepsilon^-_{tj}=y_t-\beta_{0j}-\beta_j^T x_t, && \forall t\in T,\forall j\in J,\\
&\varepsilon^+_{tj},\varepsilon^-_{tj}\geq 0, && \forall t\in T,\forall j\in J,\\
&\xi_{p\alpha}\geq \beta_{pj}, && \forall p\in P,\forall j\in J,\\
&\xi_{p\alpha}\geq -\beta_{pj}, && \forall p\in P,\forall j\in J\\
&\beta_{0,j-1}+\beta^T_{j-1}x_t\leq \beta_{0j}+\beta_j^T x_t, && \forall t\in T,\forall j\in J_{(-1)},
\end{aligned}$$

Variable Selection via LASSO

- We then define S_λ as the set of indexes of selected variables given by

$$S_\lambda = \{p \in \{1, \dots, P\} | \, |\beta^{*LASSO}_{\lambda,p}| \neq 0\}.$$

Hence, we have that, for each $p \in \{1, \dots, P\}$,

$$\beta^{*LASSO}_{\theta,p} = 0 \Longrightarrow \beta^{*}_{\theta,p} = 0.$$

- On the second stage, we estimate coefficients using a regular QR where input variables are only the ones which belonging to S_λ

MILP - Defining groups for α-quantiles

$$
\begin{aligned}
\min_{\beta_{0j},\beta_j,z_{pj},\varepsilon^+_{tj},\varepsilon^-_{tj}} \quad & \sum_{j\in J}\sum_{t\in T}\left(\alpha_j\varepsilon^+_{tj} + (1-\alpha_j)\varepsilon^-_{tj}\right) & \\
\text{s.t} \quad & \varepsilon^+_{tj} - \varepsilon^-_{tj} = y_t - \beta_{0j} - \beta_j^T x_t, & \forall t \in T, \forall j \in J, \\
& \varepsilon^+_{tj}, \varepsilon^-_{tj} \geq 0, & \forall t \in T, \forall j \in J, \\
& \mathcal{Z}_{pjg} := 2 - (1 - z_{pg}) - I_{gj}, & \\
& -M\mathcal{Z}_{pjg} \leq \beta_{pj} \leq M\mathcal{Z}_{pjg}, & \forall j \in J, \forall p \in P, \forall g \in G \\
& \sum_{p\in P} z_{pg} \leq K, & \forall j \in J, \\
& \beta_{0,j-1} + \beta_{j-1}^T x_t \leq \beta_{0j} + \beta_j^T x_t, & \forall t \in T, \forall j \in J_{(-1)}, \\
& I_{gj}, z_{pg} \in \{0,1\}, & \forall p \in P, \forall g \in G, \\
& z_{pg} \in \{0,1\}, & \forall j \in J, \forall p \in P,
\end{aligned}
$$

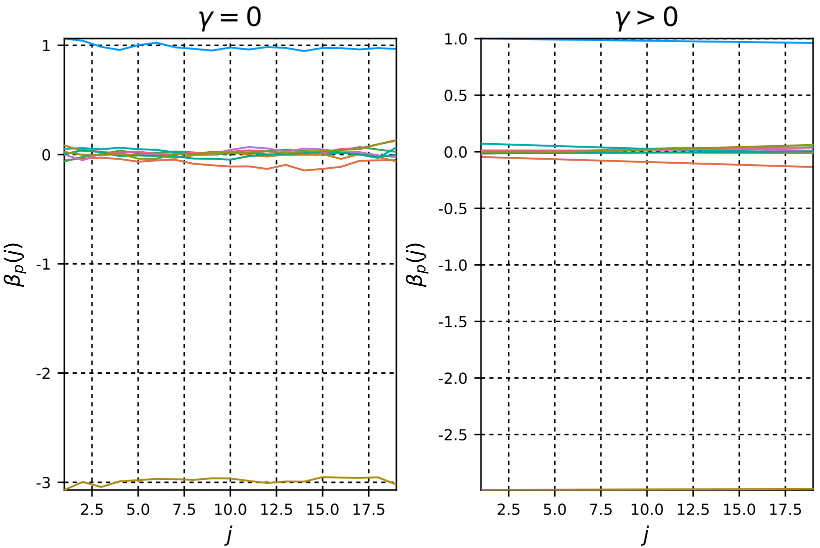
MILP - Penalization of derivative

$$
\begin{aligned}
\min_{\beta_{0j},\beta_j,z_{pj}\varepsilon^+_{tj},\varepsilon^-_{tj}} \; & \sum_{j\in J}\sum_{t\in T}\left(\alpha_j\varepsilon^+_{tj} + (1-\alpha_j)\varepsilon^-_{t\alpha}\right) + \gamma \sum_{j\in J'} D2_{pj} & \\
\text{s.t} \quad & \varepsilon^+_{tj} - \varepsilon^-_{tj} = y_t - \beta_{0j} - \beta_j^T x_t, & \forall t\in T, \forall j \in J, \\
& \varepsilon^+_{tj}, \varepsilon^-_{tj} \geq 0, & \forall t\in T, \forall j \in J, \\
& -Mz_{pj} \leq \beta_{pj} \leq Mz_{pj}, & \forall j\in J, \forall p \in P, \\
& \sum_{p\in P} z_{pj} \leq K, & \forall j \in J, \\
& z_{pj} \in \{0,1\}, & \forall j\in J, \forall p \in P, \\
& \tilde{D}^2_{pj} := \frac{\left(\frac{\beta_{p,j+1}-\beta_{pj}}{\alpha_{j+1}-\alpha_j}\right) - \left(\frac{\beta_{p,j}-\beta_{p,j-1}}{\alpha_J-\alpha_{j-1}}\right)}{\alpha_{j+1} - 2\alpha_j + \alpha_{j-1}} & \\
& D2_{pj} \geq \tilde{D}^2_{pj} & \forall j\in J_{(-1)}, \forall p \in P, \\
& D2_{pj} \geq -\tilde{D}^2_{pj} & \forall j\in J_{(-1)}, \forall p \in P, \\
& \beta_{0,j-1} + \beta_{j-1}^T x_t \leq \beta_{0j} + \beta_j^T x_t, & \forall t\in T, \forall j \in J_{(-1)},
\end{aligned}
$$

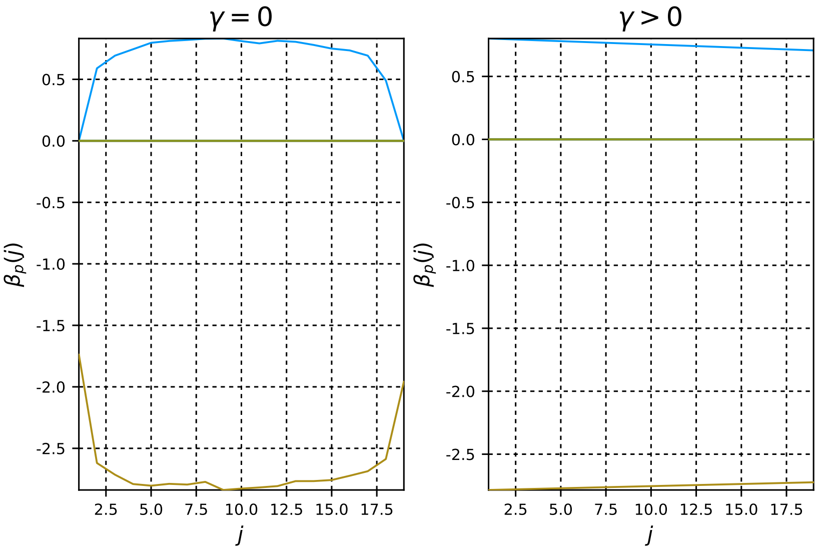
LASSO - Penalization of derivative

$$
\begin{aligned}
\min_{\beta_0,\beta,\varepsilon_{tj}^+,\varepsilon_{tj}^-} \quad & \sum_{j\in J}\sum_{t\in T}(\alpha_j\varepsilon_{tj}^+ + (1-\alpha_j)\varepsilon_{tj}^-) + \lambda\sum_{p\in P}\xi_{pj} + \gamma\sum_{j\in J'} D2_{pj} \\
\text{s.t.} \quad & \varepsilon_{tj}^+ - \varepsilon_{tj}^- = y_t - \beta_{0j} - \beta_j^T x_{t,p}, && \forall t\in T, \forall j \in J,\\
& \varepsilon_{tj}^+, \varepsilon_{tj}^- \geq 0, && \forall t\in T, \forall j \in J,\\
& \xi_{pj} \geq \beta_{pj}, && \forall p\in P, \forall j \in J,\\
& \xi_{pj} \geq -\beta_{pj}, && \forall p\in P, \forall j \in J,\\
& \tilde{D}_{pj}^2 := \frac{\left(\frac{\beta_{p,j+1}-\beta_{pj}}{\alpha_{j+1}-\alpha_j}\right) - \left(\frac{\beta_{p,j}-\beta_{p,j-1}}{\alpha_j-\alpha_{j-1}}\right)}{\alpha_{j+1}-2\alpha_j+\alpha_{j-1}} \\
& D2_{pj} \geq \tilde{D}_{pj}^2 && \forall j\in J_{(-1)}, \forall p \in P,\\
& D2_{pj} \geq -\tilde{D}_{pj}^2 && \forall j\in J_{(-1)}, \forall p \in P,\\
& \beta_{0j} + \beta_j^T x_t \leq \beta_{0,j+1} + \beta_{j+1}^T x_t, && \forall t\in T, \forall j \in J_{(-1)},
\end{aligned}
$$

LASSO - Penalization of derivative



LASSO - Penalization of derivative



LASSO - Penalization of derivative

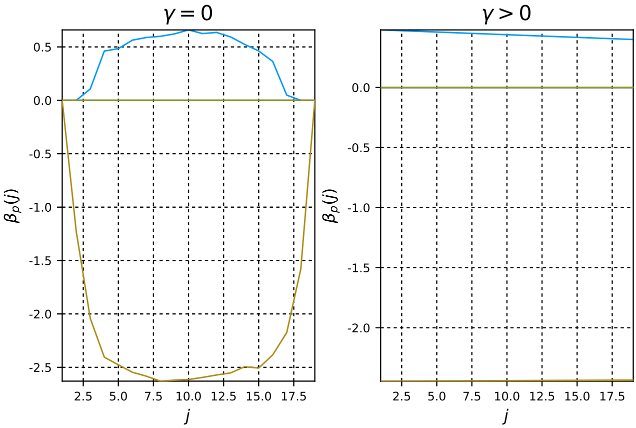


Figure: Testing caption

ADALASSO

LASSO solutions are solutions that minimize

$$Q(\beta|X,y) = \frac{1}{2n} \parallel y - X\beta \parallel^2 + \lambda \sum_{p \in P} \mid \beta_p \mid .$$

The adaptive lasso simply adds weights to this to try to counteract the known issue of LASSO estimates being biased.

$$Q_a(\beta|X,y,w) = \frac{1}{2n} \parallel y - X\beta \parallel^2 + \lambda \sum_{p \in P} w_p \mid \beta_p \mid .$$

Often you will see $w_p = 1/\tilde{\beta}_p$, where $\tilde{\beta}_p$ are some initial estimates of the β (maybe from just using LASSO, or using least squares, etc). Sometimes adaptive lasso is fit using a pathwise approach where the weight is allowed to change with λ:

$$w_p(\lambda) = w(\tilde{\beta}_{p,t}(\lambda)).$$

In the **glmnet** package the weights can be specified with the penalty.factor argument. I'm not sure if you can specify the pathwise approach in glmnet.

ADALASSO for quantiles

The problem modified for quantiles

- **First step:** Normal lasso regularization

$$\min_{\beta_{0j},\beta_j} \sum_{j\in J} \left(\sum_{t\in T} \rho_{\alpha_j}(y_t - (\beta_{0j} + \beta_j^T x_t)) + \lambda \sum_{p\in P} \mid \beta_{pj} \mid \right),$$

- **Second step:** Use initial estimation to determine w_{pj}. We can use two different approaches:
  1. $w_{pj} = 1/ \parallel \beta_j \parallel_1$,
  2. $w_{pj} = 1/\beta_{pj}$.

  The weights w_j are input to a second-stage Lasso estimation:

$$\min_{\beta_{0j},\beta_j} \sum_{j\in J} \left(\sum_{t\in T} \rho_{\alpha_j}(y_t - (\beta_{0j} + \beta_j^T x_t)) + \lambda \sum_{p\in P} w_{pj}^{\delta} \mid \beta_{pj} \mid \right),$$

  where δ is an exponential parameter, normally set to 1.

Evaluation Metrics

- We use a performance measurement which emphasizes the correctness of each quantile. For each probability $\alpha \in A$, a loss function is defined by

$$L_\alpha(q) = \sum_{t \in T} \rho_\alpha(y_t - q_\alpha(x_t)).$$

The loss score $\mathcal{L}$, which is the chosen evaluation metric to optimize, aggregates the score function over all elements of A:

$$\mathcal{L} = \frac{1}{|A|} \sum_{\alpha \in A} L_\alpha(q).$$

Time-series Cross-Validation

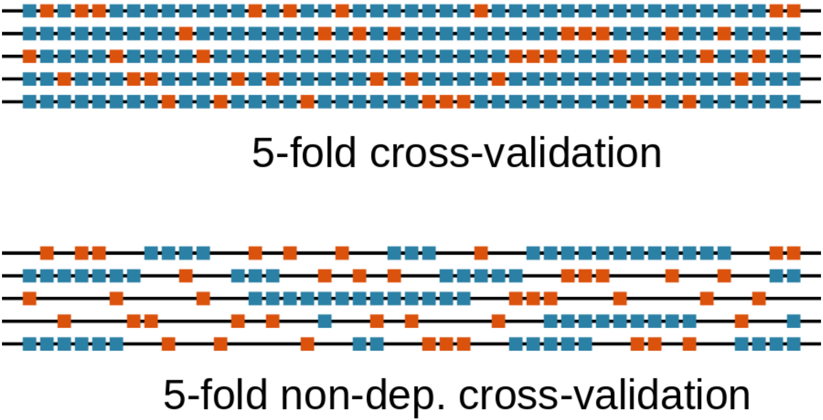


Figure: $\mathcal{K}$-fold CV and $\mathcal{K}$-fold with non-dependent data. Observations in blue are used to estimation and in orange for evaluation. Note that non-dependent data doesn't use all dataset in each fold.

Time-series Cross-Validation

- The CV score is given by the sum of the loss function for each fold. The optimum value of t in this criteria is the one that minimizes the CV score:

$$\theta^* = \text{argmin}_\theta CV(\theta) = \sum_{k \in \mathcal{K}} \sum_{\alpha \in A} L_\alpha(q).$$

- To optimize CV function in θ, we use the Nelder-Mead algorithm, which is a known and widely used algorithm for black-box optimization.

Nonparametric model

$$\hat{Q}_{Y|X}(\alpha,\cdot) \quad \in \quad \underset{q(\cdot)\in\mathcal{Q}}{\arg\min}\, L_\alpha(q) = \sum_{t\in T} \rho_\alpha(y_t - q(x_t)),$$

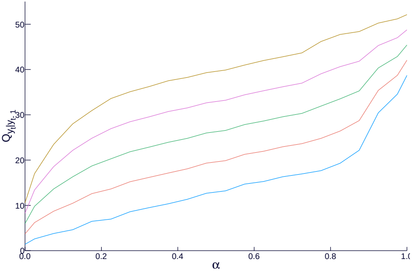
- On nonparametric models, q_α belongs to a space of limited second derivative function $\mathcal{Q}$.
- The α-quantile function is flexible enough to capture nonlinearities on the quantile function.

Nonparametric model - Formulation

$$
\begin{aligned}
\min_{q_{jt},\varepsilon_t^+,\varepsilon_t^-,\xi_t} \quad & \sum_{j\in J}\sum_{t\in T'}\left(\alpha_j\varepsilon_{tj}^+ + (1-\alpha_j)\varepsilon_{tj}^-\right) + \lambda\sum_{t\in T'}\xi_{tj} \\
s.t. \quad & \varepsilon_t^+ - \varepsilon_{tj}^- = y_t - q_{tj}, && \forall t\in T', \forall j\in J, \\
& D_{tj}^1 = \frac{q_{jt+1}-q_{jt}}{x_{t+1}-x_t}, && \forall t\in T', \forall j\in J, \\
& D_{tj}^2 := \frac{\left(\frac{q_{jt+1}-q_{jt}}{x_{t+1}-x_t}\right)-\left(\frac{q_{jt}-q_{jt-1}}{x_t-x_{t-1}}\right)}{x_{t+1}-2x_t+x_{t-1}} \\
& \xi_{tj} \geq D_{tj}^2, && \forall t\in T', \forall j\in J, \\
& \xi_{tj} \geq -D_{tj}^2, && \forall t\in T', \forall j\in J, \\
& \varepsilon_{tj}^+, \varepsilon_{tj}^-, \xi_{tj} \geq 0, && \forall t\in T', \forall j\in J, \\
& q_{tj} \leq q_{t,j+1}, && \forall t\in T', \forall j\in J,
\end{aligned}
$$

Nonparametric vs. Linear Model

- The nonparametric approach is more flexible to capture heteroscedasticity.



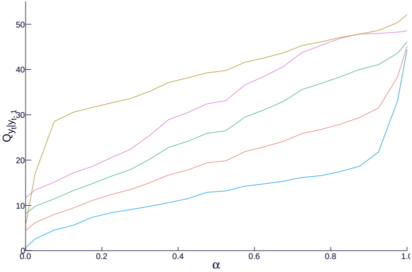
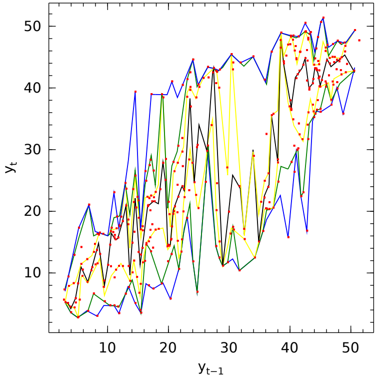


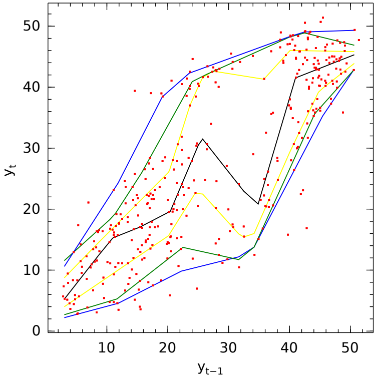
Figure: Estimated quantile functions, for different values of y_{t-1}. On the left using a linear model and using a nonparametric approach on the right.

Control of smoothing parameter

- This flexibility might lead to overfitting, if we don't select a proper smoothing parameter.

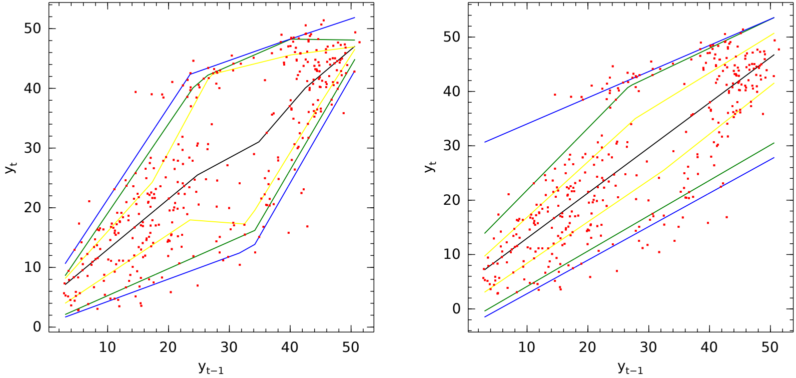


(a) $\lambda = 0.1$



(b) $\lambda = 3$

Control of smoothing parameter



(a) $\lambda = 10$ (b) $\lambda = 100$

- On the limit, when $\lambda \to \infty$, the nonparametric model approaches a linear model.

Present issues

- Difficult interpolation when x_t has dimension greater than 1.
- Control of smoothing parameter

References

Gallego-Castillo, Cristobal, Ricardo Bessa, Laura Cavalcante, and Oscar Lopez-Garcia. 2016. "On-Line Quantile Regression in the Rkhs (Reproducing Kernel Hilbert Space) for Operational Probabilistic Forecasting of Wind Power." *Energy* 113. Elsevier: 355–65.

Koenker, Roger, and Gilbert Bassett Jr. 1978. "Regression Quantiles." *Econometrica: Journal of the Econometric Society*. JSTOR, 33–50.

Wan, C., J. Lin, J. Wang, Y. Song, and Z. Y. Dong. 2017. "Direct Quantile Regression for Nonparametric Probabilistic Forecasting of Wind Power Generation." *IEEE Transactions on Power Systems* 32 (4): 2767–78. doi:10.1109/TPWRS.2016.2625101.