Semi-parametric models and robust aggregation for GEFCom2014 probabilistic electric load and electricity price forecasting

Pierre Gaillard[a,b], Yannig Goude[a], Raphaël Nedellec[a]

[a] *EDF R&D, Clamart, France*
[b] *GREGHEC: HEC Paris – CNRS, Jouy-en-Josas, France*

Abstract

We sum up the methodology of the team Tololo who ranks 1^{st} on the load forecasting track and the price forecasting track of the Global Energy Forecasting Competition 2014. During the competition, we used and tested many statistical and machine learning methods such as random forests, gradient boosting machines and generalized additive models. In this paper, we only present the methods that have shown the best results. For electric load forecasting, our strategy consists in producing temperature scenarios that we plug into a probabilistic forecasting load model. Both steps are performed by fitting a quantile generalized additive model (quantGAM). Concerning the electricity price forecasting, we investigate three methods that we used during the competition. The first method follows the spirit of the one used for electric load. The second one is based on combining a set of individual predictors. The last one fits a sparse linear regression on a large set of covariates. We chose to present in this paper these three methods because they show good performance and have a nice potential of improvements for future research.

Keywords: electric load, electricity price, forecast, semi-parametric models, combining forecasts, sparse regression

1. Introduction

We present in this paper the methodology employed for the probabilistic electric load and electricity price forecasting tracks of the Global Energy Forecasting Competition 2014 (GEFCom2014). We participated in both tracks but with different intensity and motivation. Load forecasting was a familiar field of research for us before the competition, whereas we were inexperienced with price forecasting. As a consequence, we converged rapidly to a unique solution for load forecasting, but we constantly changed our method as we were learning and improving our knowledge of electricity price forecasting.

Quantile regression based on pinball loss minimization (see Koenker and Bassett 1978) and generalized additive models (see Hastie and Tibshirani 1990; Wood 2006) are the main tools of our work. To our knowledge, there was no off-the-shelf program achieving quantile generalized additive models and we implemented our own solution for that. We designed it originally for load forecasting but at the end it turned out to be the most efficient method for both tasks. We present it in Section 2. We tested a wide range of other approaches for the price forecasting task. Among those we describe those which deserve to be shared. In our opinion, they have a potential for improvement and can be applied to other forecasting problems. Aggregation of experts is considered in Section 4.3. We were inspired by the work of Nowotarski and Weron (2014) and extend it to the case where the weights of the combination can vary over time. More precisely, we adapt to quantile regression the setting of robust online aggregation of experts (see Cesa-Bianchi and Lugosi 2006) which has already been applied successively for point wise load forecasting in Devaine et al. (2013) and Gaillard and Goude (2014). Our set of 13 experts consists of forecasters from the price forecasting literature AR (autoregressive models), TAR (treshold autoregressive), ARX (autoregressive exogenous), TARX (treshold autoregressive exogenous), PAR (spike preprocessed autoregressive) as presented in Weron and Misiorek 2008), GAMs (generalized additive models), random forest (see Breiman 2001) and gradient boosting machines (see Friedman 1999). The third approach presented in Section 4.4 is based on covariate selection with ℓ_1 selection procedure, commonly known as Lasso regression introduced in Tibshirani (1996). It was motivated by the fact that we generated a lot of covariates (192) from the original ones (for price forecasting). Thus, we were curious at the end of the competition to see how an automatic procedure could select an optimal subset among them. ℓ_1 selection and quantile regression were studied in Belloni and Chernozhukov (2011) but never applied neither to price nor to load forecasting. To our experience, no open source code exists that satisfies our needs for the competition. We present in Section 4.4 a kernel based approach we developed at this occasion. For the price forecasting task, the results obtained during the competition differ slightly (sometimes better, sometimes worse) from those obtained by the three methods (quantile GAM, quantile mixture, quantile GLM (generalized linear model)). This is largely due to the other approaches that we used along the competition, and to hybrid variants of those presented here. We deliberately focus on these three methods in this paper for conciseness.

Email addresses: pierre-p.gaillard@edf.fr (Pierre Gaillard), yannig.goude@edf.fr (Yannig Goude), raphael.nedellec@edf.fr (Raphaël Nedellec)

2. Quantile regression with Generalized Additive Models

We consider the supervised regression setting where we are asked to forecast an univariate response variable $Y_t \in \mathbb{R}$ (such as the load) according to several covariates $\boldsymbol{X}_t = (X_{t,1}, \ldots, X_{t,d}) \in \mathbb{R}^d$ (such as the temperature). A training sample $\{(\boldsymbol{X}_t, Y_t)\}_{t=1}^n$ is available.

2.1. Generalized Additive Models

GAMs were introduced by Hastie and Tibshirani (1990). GAMs explain the output Y_t as a sum of smooth functions of the different covariates $X_{t,j}$. More formally, we assume that for all time $t = 1, \ldots, n$, $Y_t = \mu(\boldsymbol{X}_t) + \varepsilon_t$ where μ is the unknown function to be estimated and the ε_t denote zero mean random variables that are independent and identically distributed (i.i.d.) from some exponential family distribution[1]. GAMs assume that it exists a link function g such that

$$g(\mu(\boldsymbol{X}_t)) = f_1(X_{t,1}) + f_2(X_{t,2}) + f_3(X_{t,3}, X_{t,4}) + \ldots \quad (1)$$

where the f_j are smooth functions of the covariates $X_{t,k} \in \mathbb{R}$. In the following, the link function g is the identity and the smooth functions f_j are cubic splines (unless specified otherwise). Basically, cubic splines are polynomials of degree 3 that are joined at points known as "knots" by satisfying some continuity constraints (see Wood 2006 for details). We call $\mathcal{S}(K_i)$ the class of cubic splines for some fixed set K_i of knots.

We fit the smooth effects f_i with penalized regression methods. To do so, we first choose the knots K_i for each effect f_i. Then, we use the ridge regression that minimizes over all effects $f_1 \in \mathcal{S}(K_1), f_2 \in \mathcal{S}(K_2), \ldots$ the following criterion:

$$\sum_{t=1}^{n} \left(Y_t - \sum_{i=1}^{p} f_i\left(X_t^i\right) \right)^2 + \sum_{i=1}^{p} \lambda_i \int \left\| f_i''(x) \right\|_2^2 dx, \quad (2)$$

where for each effect X_t^i are one or two covariates of $\boldsymbol{X}_t$ corresponding to effect f_i. Here $\lambda_1, ..., \lambda_p > 0$ are regularization parameters that control the degree of smoothness of each effect (the higher λ_i the smoother f_i is). They have to be optimized. The knots K_i are uniformly distributed over the range of the covariate(s) X_t^i corresponding to effect f_i. The number of knots (i.e., the cardinal of K_i) is another way to control the smoothness of the effect f_i and should be optimized as well. These problems are solved by using the methodology presented in Wood (2006) which consists in minimizing the Generalized Cross Validation criterion (GCV). The method is implemented in the R package `mgcv` (see Wood, 2006).

2.2. Quantile regression

Quantile regression was introduced by Koenker and Bassett (1978). Let Y be a real value random variable and let $\boldsymbol{X}$ be a set of explanatory variables. If $F_{Y|\boldsymbol{X}}$ denotes the conditional cumulative distribution of Y given $\boldsymbol{X}$, then the conditional quantile q_τ of order $\tau \in [0, 1]$ of Y knowing X is defined as the generalized inverse of $F_{Y|\mathbf{X}}$:

$$q_\tau(Y|\boldsymbol{X}) = F^{-1}_{Y|\boldsymbol{X}}(\tau) = \inf \{ y \in \mathbb{R}, F_{Y|\boldsymbol{X}}(y) \geq \tau \}. \quad (3)$$

Now, the idea of quantile estimation arises from the observation that the median (i.e., $q_{0.5}(Y|\boldsymbol{X})$) minimizes the expected absolute error. More generally, it can be shown that the conditional quantile $q_\tau(Y|\boldsymbol{X})$ is the solution of the minimization problem:

$$q_\tau(Y|\boldsymbol{X}) \in \arg\min_g \mathbb{E}[\rho_\tau(Y - g(\boldsymbol{X}))|\boldsymbol{X}], \quad (4)$$

where ρ_τ is the pinball loss defined for all $u \in \mathbb{R}$ by $\rho_\tau(u) = u(\tau - \mathbf{1}_{\{u<0\}})$.

Linear quantile regression is implemented in the R-package `quantreg` (see Koenker 2013). It assumes that $\{(\mathbf{X}_t, Y_t)\}_{t=1,\ldots,n}$ are i.i.d. such that $Y_t = \mathbf{X_t}^\top \boldsymbol{\beta} + \varepsilon_t$, where $\boldsymbol{\beta} \in \mathbb{R}^d$ is a vector of unknown parameters. Linear quantile regression solves the convex minimization problem

$$\widehat{\boldsymbol{\beta}}_\tau \in \arg\min_{\boldsymbol{\beta} \in \mathbb{R}^d} \sum_{t=1}^{n} \rho_\tau(Y_t - \mathbf{X}_t^\top \boldsymbol{\beta}), \quad (5)$$

and estimates q_τ with $\widehat{q_\tau} : \boldsymbol{x} \mapsto \boldsymbol{x}^\top \widehat{\boldsymbol{\beta}}_\tau$.

2.3. Mixed approach: use smooth effects estimated by GAM as input for linear quantile regression

In this section, we introduce a generic procedure to perform quantile regression by using GAMs. An obvious patch would be to substitute in (2) the square loss with the pinball loss. However, the optimization problem is harder and we faced numerical problems trying to solve it. During the competition, we used a multiple steps approach to deal with these computational problems. We detail the methodology, called quantGAM, below.

i) We linearize the problem by fitting GAM. To do so, we perform the two following steps.
   - *Fit the mean.* We fit GAM by minimizing Criterion (2) over the non-linear effects $f_1 \in S(K_1), f_2 \in S(K_2), \ldots$. We get estimates of the effects (denoted $\widehat{f_i}$ for all i) that capture the non-linear relationships between the conditional mean of Y_t and the covariates. We denote by $\widehat{Y_t}$ the fitted estimate of observation Y_t formed by GAM.
   - *Fit the variance* (optional). We perform a residual analysis to model the deviations from the mean knowing the covariates. To do so, we fit GAM to predict the square residuals $(Y_t - \widehat{Y_t})^2$. We obtain estimates of the effects (denoted $\widehat{g_i}$) that trap the second-order variation of Y_t. This step is optional.

i)) Finally, for each $\tau \in \{0.01, 0.02, ..., 0.99\}$, we perform a linear quantile regression by using the estimated effects $\boldsymbol{Z}_t = (\widehat{f_1}(X_{t,1}), \widehat{f_2}(X_{t,2}), \ldots, \widehat{g_1}(X_{t,1}), \widehat{g_2}(X_{t,2}), \ldots)$ as covariates to estimate the quantile function $\widehat{q_\tau}$. To do so, we substitute in (5) the vector of covariates $\boldsymbol{X}_t$ with the vector of fitted effect $\boldsymbol{Z}_t$ and solve the minimization problem.

[1] Throughout the paper, ε_t are i.i.d. error terms but their distribution may change from a display to another

3. Probabilistic electric load forecasting by quantGAM

We consider the data available for the Probabilistic Load Forecasting Track of the GEFCom2014 competition. It includes hourly observations of load consumption (from January 1, 2006 to December 31, 2011) and of temperatures from twenty-five weather stations (from January 1, 2001 to December 31, 2011). Our goal is to forecast at the end of each month the τ-quantiles of the load consumption of the next month for $\tau \in \{0.01, \ldots, 0.99\}$. At time step t, the performance of a prediction $(\widehat{q}_{0.01,t}, \ldots, \widehat{q}_{0.99,t}) \in \mathbb{R}^{99}$ is measured by the average pinball loss over the quantiles defined as:

$$(1/99) \sum_{\tau=1}^{99} \rho_\tau(Y_t - \widehat{q}_{\tau,t}) \,. \tag{6}$$

The data set is parsed into two pieces: a training set from 2001 to 2010 and a testing set in 2011. The testing set is predicted online (month by month) by fitting the methods on all the past observations (including the beginning of the testing set). The electric demand and the temperature heavily depend on the hour of the day. Therefore the models described throughout this section are performed per hour (unless stated otherwise). That is, the data is partitioned into 24 independent time series (one for each hour of the day) and 24 separate models are fitted.

The importance of the past consumptions and temperatures is clearly decreasing over time. This has driven us to use two approaches depending on the forecasting horizon: one taking into account these dependencies for "short-term" load forecasting and a second approach for "mid-term" probabilistic load forecasting.

Section 3.4 reports the performance of the method obtained for each month of the testing set.

3.1. Data preprocessing

These are the variables that have been defined in order to build the forecasting models and methods:

- Y_t is the electric load at time $t \geq 1$[2].
- T_t is a uniform weighted average of the temperatures of the weather stations 6, 10, 22, and 25. We choose these stations with a relatively simple method. Considering the simplified hourly GAM of equation (10), we test successively the impact of each temperature station. We choose these four stations using generalized cross validation scores. We represent on Figure 1 the GCV score obtained for each temperature station compared to the one obtained by T_t. We clearly see that the 4 selected stations have similar GCV scores and that averaging them brings a significant improvement.
- $T_t^{(\gamma)}$ is a smoothed temperature of T_t with exponential smoothing parameter $\gamma \in [0, 1]$. It is defined at time t by induction as:

$$T_t^{(\gamma)} \triangleq \gamma \, T_{t-1}^{(\gamma)} + (1 - \gamma) \, T_t \,. \tag{7}$$

[2]Throughout the paper, $t \geq 1$ is a linear chronological index (in hours) for the whole dataset.

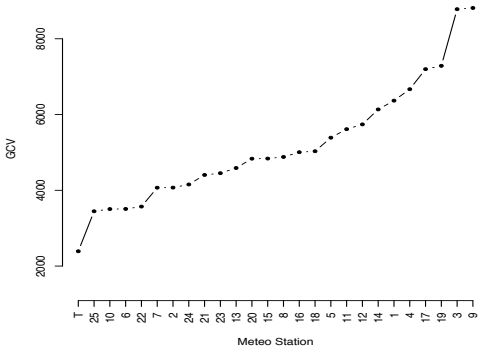

Figure 1: GCV score obtained for each temperature station compared to the one obtained by the average temperature T.

- $Toy_t \in [0, 1]$ (Time of year) is a cyclic variable that indicates the annual position and repeats each year. It is each year linearly increasing over time going from 0 on January 1 at 00:00 to 1 on December 31 at 23:30.
- $DayType_t$ is a factorial variable with 7 levels corresponding to different types of day. The levels are: Monday, Tuesday-Wednesday-Thursday, Friday, Saturday, Sunday, bank holidays, and a last category corresponding to the days before and after bank holidays. This choice was driven by our expertise on electricity load data (see e.g., Goude et al., 2012).

3.2. Medium-term probabilistic forecasting of the load

To forecast the electric demand at more than two days horizons, we use a medium term probabilistic model that does not use recent lags of the load. In fact, we validated (by performing cross validation) the correct horizon where recent observations of the temperature (by using scenarios) were still informative. It appears that the right horizon was around 48 hours.

We separate the uncertainty of the model and the uncertainty due to the temperature. First we build a medium-term probabilistic forecasting model of the temperature only based on the impact of the annual position Toy_t. Then we fit a model that forecasts the distribution of the load conditionally to the temperature (assuming that true temperature is known in advance). Both models are performed by using quantGAM described in Section 2.3. We form our final prediction of the load distribution by averaging the forecasted conditional distribution of the load over the forecasted law of the temperature.

Probabilistic forecasting of the temperature. We perform quantGAM, by following the steps of the generic procedure of Section 2.3, as follows:

i) We estimate the non-linear effect of the annual position Toy_t on the expected temperature by fitting GAM (i.e., by minimizing Criterion (2) substituting the response variable Y_t with the temperature T_t) with the following model:

$$T_t = f_1(Toy_t) + \varepsilon_t \,. \tag{8}$$

Here f_1 is estimated by cubic cyclic regression splines to assure continuity of estimated effects at midnight the 1st of

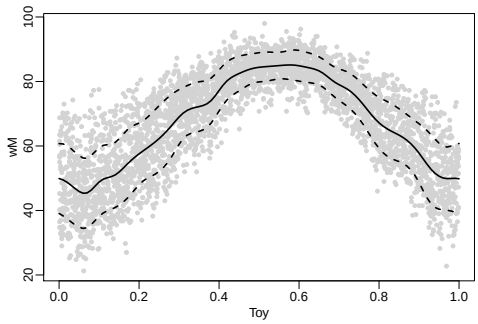

Figure 2: Observed values of T_t together with the smooth functions $\widehat{f_1}$ and $\widehat{f_1} \pm \widehat{g_1}$ fitted by Models (8) and (9).

January (see Wood 2006). We denote by $\widehat{f_1}$ the obtained estimate of f_1 (i.e., the solution of the minimization of (2)).

ii) We estimate the non-linear effect that impacts the forecasting errors by fitting GAM on the residual signal with model:

$$\left(T_t - \widehat{f_1}(Toy_t)\right)^2 = g_1(Toy_t) + \varepsilon_t\,, \tag{9}$$

where g_1 is estimated by cubic cyclic regression splines. We call $\widehat{g_1}$ the estimate of g_1.

iii) We perform linear quantile regression (see Section 2.2) to forecast the quantiles of the temperature by using

$$\boldsymbol{Z_t} = \left(\widehat{f_1}(Toy_t),\, \widehat{g_1}(Toy_t)\right)$$

as vector of covariates. The final estimates of the τ-quantiles, denoted $\widehat{T}_{\tau,t}$, are thus linear combinations of the mean effect $\widehat{f_1}$ and the variance effect $\widehat{g_1}$. That is, they are of the form:

$$\widehat{T}_{\tau,t} = \widehat{a}_{\tau,1}\,\widehat{f_1}(Toy_t) + \widehat{a}_{\tau,2}\,\widehat{g_1}(Toy_t)\,,$$

where $\widehat{a}_{\tau,1}, \widehat{a}_{\tau,2} \in \mathbb{R}$ are the linear coefficients estimated by the quantile regression.

Figure 2 plots the estimates $\widehat{f_1}$ and $\widehat{f_1} \pm \widehat{g_1}$ together with the observed values of the temperatures T_t.

Probabilistic forecasting of the load (knowing the temperature in advance). We fit quantGAM as follows:

i) We estimate the non-linear effects that impact the expected load by fitting GAM with model:

$$Y_t = f_1(Toy_t) + f_2(t) + f_3(T_t) + h(DayType_t) + \varepsilon_t\,, \tag{10}$$

where f_1, f_2, and f_3 are cubic regression splines and h is a function that takes a different value for each type of day. We recall that the estimation of GAM is performed by minimizing (2) over all cubic splines $f_i \in S(K_i)$ and over all $h \in \{\text{Monday}, \dots\}^{\mathbb{R}}$. Note that h actually corresponds to seven additional coefficients that do not appear in the regularization term of (2). Note that f_2 captures the trend.

ii) We estimate the non-linear effects that impact the forecasting errors by fitting GAM on the residual signal with model:

$$\left(Y_t - \widehat{Y_t}\right)^2 = g_1(Toy_t) + g_2(T_t) + \varepsilon_t\,, \tag{11}$$

where $\widehat{Y_t}$ is the load fitted by Model (10).

iii) We perform linear quantile regression to forecast the quantiles of the load by using the covariate vector

$$\boldsymbol{Z}_t = (\widehat{f_1}(Toy_t),\, \widehat{f_2}(t),\, \dots\,,\, \widehat{g_1}(Toy_t),\, \widehat{g_2}(T_t))\,.$$

Figure 3 plots the forecasted distribution of the load for three consecutive days by using the observed values of T_t. The forecasted distribution does not take into account the uncertainty due to the temperature and the obtained confidence intervals are thus extremely small.

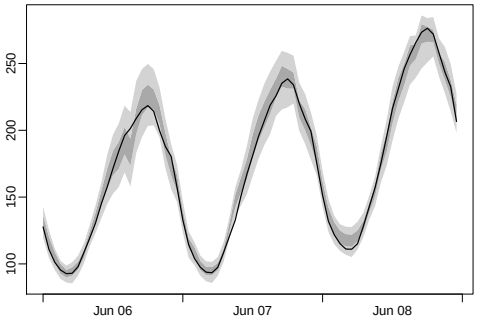

Figure 3: Medium-term forecasted distribution of the load from June 6, 2011 to June 8, 2011 by using the real values of the temperature T_t. As in all the following plots of probabilistic distributions, we only plot the 50% confidence interval in dark gray and the 90% confidence interval in light gray.

Probabilistic forecasting of the load (operational forecast). To provide forecasts of the load distribution, we do not have at our disposal the future true values of the temperature T_t. To deal with it, we average the forecasted distributions of the load over the forecasted distribution of temperature. In other words, to generate quantile forecasts of the load at each time step t:

i) We forecast all the quantiles $\tau \in \{0.01, \dots, 0.99\}$ of the temperature at time step t by quantGAM described earlier (see Equations (8) and (9)).

ii) For each predicted quantile $\widehat{T}_{\tau,t}$ of the temperature, we perform quantGAM (described in Models (10) and (11)) by substituting the true value of T_t with the predicted quantile $\widehat{T}_{\tau,t}$ and we obtain a distribution $\widehat{F}_{\tau,t}$ of the load.

iii) We form the final prediction of the load distribution by averaging over all temperature percentiles τ the forecasted distribution $\widehat{F}_t = (1/99)\sum_{\tau=1}^{99} \widehat{F}_{\tau,t}$. In the end for each level $\tau' \in \{0.01, \dots, 0.99\}$ we predict the percentile of the load $\widehat{q}_{\tau',t} = \widehat{F}_t^{-1}(\tau')$ by inverting the forecasted distribution $\widehat{F}_t$.

Figure 4 plots the forecasted distribution for the same three consecutive days as for Figure 3. We remark that the obtained confidence intervals are much wider and less accurate than in Figure 3 which knew the true values of T_t.

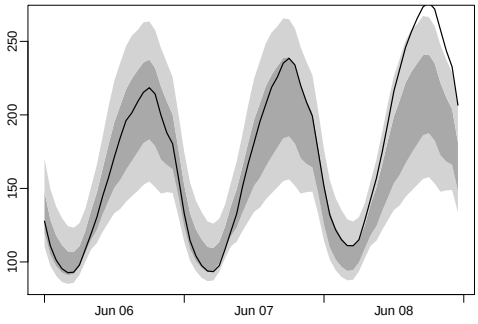

Figure 4: Medium-term forecasted distribution of the load from June 6, 2011 to June 8, 2011 obtained by averaging the forecasted distributions of the load over the forecasted distribution of the temperature.

3.3. Forecasting up to two days' horizon.

In order to forecast at a "short-term" horizon (up to two days ahead), we gain in accuracy by considering recent lag of the temperature. We thus built another method for this purpose based on Monte Carlo methods.

Basically, similarly to the medium-term model this method partitions the analysis into two different layers. Both of them use the quantGAM method described in Section 2.3. First, we generate 800 temperature scenarios by sampling step by step (one hour ahead) the next value of the temperature. Second, we plug the temperature scenarios into a probabilistic forecasting model of the load that was fitted with the true values of the temperature as exogenous variable. The final prediction of the load distribution is obtained by averaging the forecasted distributions over the 800 simulated scenarios.

We explain below how the temperature scenarios are generated.

Generating randomly the temperature scenarios. We generate the temperature scenarios (for T_t and for the smoothed temperatures $T_t^{(0.8)}$ and $T_t^{(0.95)}$) as follows. First, we remove the annual seasonality by estimating the medium-term Model (8). Second, we consider the residual signal $e_t \triangleq T_t - \widehat{f_1}(Toy_t)$. Finally, we fit quantGAM so as to predict the distribution of the next residual temperature (one hour ahead):

i) We fit the expected residuals e_t according to Model (12) to take into account autocorrelation within the data:

$$e_t = \alpha_1 e_{t-1} + \alpha_2 e_{t-2} + \ldots + \alpha_{48} e_{t-48} + \varepsilon_t \,. \tag{12}$$

Here, α_i are coefficients to be estimated. Contrary to the other models, this residual analysis is not performed per hour.

ii) Then, we estimate the non-linear seasonality of the square error of the model. Indeed, we observed that the variance was subject to the annual position (Toy_t). Thus, we consider the fitted errors $(e_t - \widehat{e_t})^2$ and we fit GAM with model:

$$\left(e_t - \widehat{e_t}\right)^2 = g_1(Toy_t) + \varepsilon_t \,, \tag{13}$$

where g_1 is estimated by cubic cyclic regression splines.

iii) We run a linear quantile regression (presented in Section 2.2) for the quantiles $\tau \in \{0.01, \ldots, 0.99\}$ with covariates

$$\boldsymbol{Z}_t = \left(\widehat{\alpha}_1 e_{t-1},\, \widehat{\alpha}_2 e_{t-2},\, \ldots,\, \widehat{\alpha}_{48} e_{t-48},\, \widehat{g}_1(Toy_t)\right) \in \mathbb{R}^{49} \,.$$

Here $\widehat{\alpha_i}$ are the coefficients estimated in Step (i) and $\widehat{g}_1$ is the annual effect estimated by Model (13). We denote by $\widehat{e}_{\tau,t}$ the one hour ahead predicted τ-quantiles of the residual e_t.

To generate a temperature scenario for the next 48 hours, we perform sequentially (step by step) for time $\in \{t+1, \ldots, t+48\}$ the following procedure: we forecast the next τ-quantiles $\widehat{e}_{\tau,s}$ for all $\tau \in \{0.01, \ldots, 0.99\}$; we sample the next residual e_s uniformly over the set of percentiles $\widehat{e}_{\tau,s}$; we compute the next temperature $T_s = \widehat{f_1}(Toy_s) + e_s$; we move to step $s+1$. The values of $T_s^{(0.8)}$ and $T_s^{(0.95)}$ are computed from the scenario of T_s by using Definition (7).

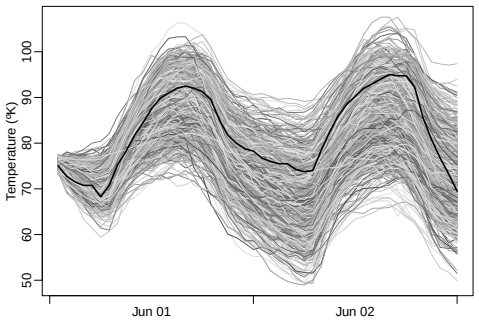

Figure 5: 800 temperature scenarios (T_s) generated for June 1, 2011 to June 2, 2011. The line in black depicts the observed temperature.

We chose to generate 800 scenarios of the temperatures. This has proven to be fast enough and to provide a good overview of what is possible. The generation of the temperature scenarios includes randomness into our method. This partly explains the slight differences into the performance reported in Table 1. Figure 5 plots the 800 scenarios simulated for June 1, 2011 to June 2, 2011 together with the observed temperature.

Probabilistic forecasting of the electric load (knowing the temperature in advance). Now we model the distribution of the load as if we had access to the true values of the temperature in advance. Once again, it is performed by fitting quantGAM described in Section 2.3. We train a separate model for each hour of the day $h = 1, \ldots, 24$ (i.e, the times series is partitioned into 24 times series that we investigate independently). The model (Step i) in Section 2.3) is defined as:

$$\begin{aligned} Y_t = f_1(T_t, t) + f_2(T_t^{(0.8)}) + f_3(T_t^{(0.95)}) \\ + f_4(Toy_t) + f_5(t) + h(DayType_t) + \varepsilon_t \,. \end{aligned} \tag{14}$$

We recall that $f_1, f_2, \ldots$ are smooth cubic splines to be estimated by GAM and that h a real function. Then, we move to Step 2 of quantGAM by performing linear quantile regressions

for all quantiles $\tau \in \{0.01, \ldots, 0.99\}$ (exceptionally, we skip Step ii)). Thus, we fit 2 376 (= 24 hours $\times$ 99 quantiles) linear quantile regressions.

Probabilistic forecasting of the electric load (operational forecast). However, to produce forecasts, once again we do not have access to the real values of the temperatures (T_t, $T_t^{(0.8)}$, and $T_t^{(0.95)}$). We need to substitute them with the 800 sampled scenarios of the temperatures. We obtain a probabilistic forecast of the load for each scenario and we form our prediction by averaging the forecasted distributions over all scenarios.

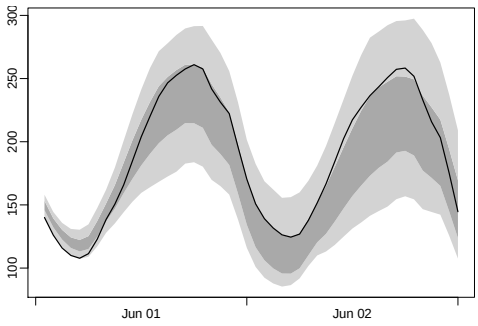

Figure 6: Forecasted distribution of the load from June 1, 2011 to June 2, 2011 by averaging the distributions obtained for each individual scenario of the temperature (T_t, $T_t^{(0.8)}$, $T_t^{(0.95)}$).

This forecasted distribution takes into account both the uncertainty with regard to the weather (captured by the temperature scenarios) and the uncertainty about the model (captured by quantGAM). Figure 6 displays the forecasted distribution for June 1, and June 2, 2011. We see that the predicted confidence intervals are tight a few hours ahead (which was not the case for the medium term forecasts displayed in Figure 4) and is widening with the horizon of prediction.

3.4. Final forecasts and results

In the end we form our forecasts for the next month by concatenating the short term forecasts (from 1 hour to 48 hours ahead) with the medium term forecasts (from 49 hours ahead).

Figure 7 shows the probabilistic forecasts obtained for the month of June 2011. Figure 8 plots the percentage of time the observed electricity consumption Y_t is smaller than the predicted quantiles $\widehat{q}_{\tau,t}$ according to $\tau \in [0, 1]$ during the year 2011. The closer the curve is from the identity function the better. We remark that the method overestimated the consumption for most quantiles. This can partly be explained by an unexpected drop of consumptions (e.g., end of August or mid-September).

Table 1 reports for each month of the year 2011 the performance obtained by the benchmark, the team Tololo, and the described methodology (used from March to December 2011 during the competition).

Table 1: Performance of the benchmark, the team Tololo, and the method described in this paper (quantGAM).

Month	Benchmark	Tololo	quantGAM
Jan.	18.74	10.44	10.62
Feb.	22.76	12.52	9.83
Mar.	13.22	8.27	7.90
Apr.	8.36	4.42	4.19
May.	10.92	5.90	5.87
Jun.	16.99	6.19	5.80
Jul.	13.40	7.32	8.13
Aug.	17.32	10.80	10.73
Sep.	13.84	5.45	5.46
Oct.	6.42	3.96	3.98
Nov.	10.94	6.32	6.33
Dec.	34.07	8.48	8.51

4. Probabilistic electricity price forecasting

We turn to the problem of electricity price forecasting. We consider the data available for the Probabilistic Electricity Price Forecasting Track of the GEFCom2014 competition. It contains hourly observations of the historical prices from January 1, 2011 to December 17, 2013, together with hourly historical zonal and system load forecasts. Our goal is to forecast one day ahead (i.e., the next 24 hours) the τ-quantiles of the electricity prices. Our performance is measured by the average pinball scored defined in (6), see Hong et al. (2015) for more details.

4.1. Data preprocessing

In order to build our forecasting method, we defined below several covariates:

- P_t is the price at time t.
- $P_t^{(\text{last})}$ is the most recent lagged price available for the forecast at time t.
- FZL_t and FTL_t are respectively the forecasted zonal and total load.
- $FZL_t^{(\gamma)}$ and $FTL_t^{(\gamma)}$ are exponential smoothing of the forecasted zonal (resp. total) load with parameter $\gamma \in [0, 1]$ (see Equation (7) for a definition).
- $X_t^{(\max)}$ (respectively $X_t^{(\min)}$ and $X_t^{(\text{mean})}$) is the maximum (respectively minimum and mean) of the variable X_t (such as the price P_t or the forecasted zonal load FZL_t) during the day corresponding to observation t.

In the sequel, we will mostly use the logarithms of the above variables.

4.2. quantGAM

We describe in this section a methodology based on quantGAM. Similarly to the electricity consumption, the electricity price heavily depends on the hour of the day and the models are hourly performed by partitioning the data into 24 independent time series.

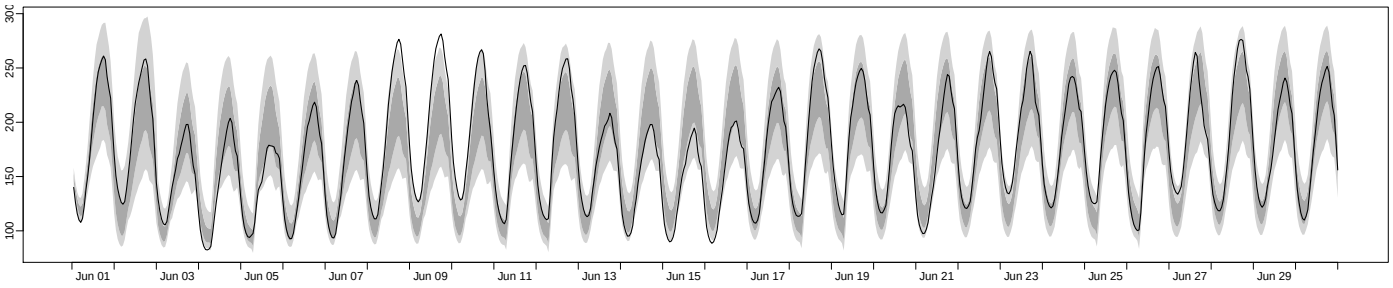

Figure 7: Forecasted distribution of the electric load from June 1, 2011 to June 30, 2011.

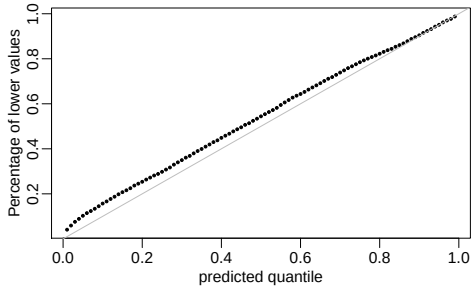

Figure 8: Percentage of observed electric load values Y_t under the predicted quantiles $\widehat{q}_{\tau,t}$ during the year 2011 which has been forecasted month by month.

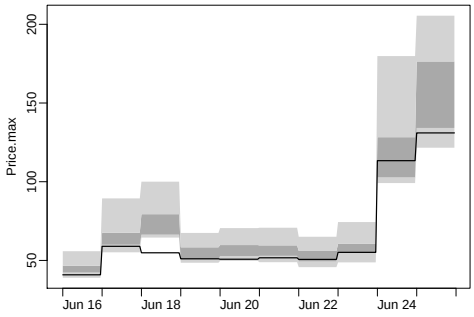

Figure 9: One day ahead forecasted distribution of the maximal price from June 16, 2011 to June 25, 2013.

4.2.1. *Probabilistic forecasting of the maximal price*

In order to predict the electricity price of the following day, we aim at using the maximal price $P_t^{(\max)}$ of the day to be predicted as an exogenous variable. To do so we should first provide a forecast of $P_t^{(\max)}$. We describe below how to do so by using quantGAM:

i). We estimate the linear and non-linear effects that impact the expected maximal price by fitting the generalized additive model:

$$\begin{aligned}\log(P_t^{(\max)}) &= \alpha_1 \log(P_{t-24}^{(\max)}) + \alpha_2 \log(P_{t-48}^{(\max)}) \\ &+ \alpha_3 \log(P_{t-24}^{(\text{mean})}) + \alpha_4 \log(P_{t-48}^{(\text{mean})}) \\ &+ f_1(\log(FTL_{t-24}^{(\text{mean})})) + f_2(\log(FTL_t^{(\text{mean})})) \\ &+ f_3(\log(FZL_t^{(\max)})) + f_4(FZL_{t-24}^{(\max)}) + \varepsilon_t\,, \qquad (15)\end{aligned}$$

where α_i are linear coefficients and f_j are smooth cubic splines to be estimated by $\widehat{\alpha}_i$ and $\widehat{f}_j$.

ii) We perform linear quantile regression to forecast the quantiles of the maximal price by using the vector of covariates

$$\begin{aligned}\boldsymbol{Z}_t = \big(&\widehat{\alpha}_1 \log(P_{t-24}^{(\max)}),\ \widehat{\alpha}_2 \log(P_{t-48}^{(\max)}), \\ &\widehat{\alpha}_3 \log(P_{t-24}^{(\text{mean})}),\ \widehat{\alpha}_4 \log(P_{t-48}^{(\text{mean})}), \\ &\widehat{f}_1(\log(FTL_{t-24}^{(\text{mean})})),\ \widehat{f}_2(\log(FTL_t^{(\text{mean})})), \\ &\widehat{f}_3(\log(FZL_t^{(\max)})),\ \widehat{f}_4(FZL_{t-24}^{(\max)})\big)\,.\end{aligned}$$

Figure 9 plots the forecasted distribution of the maximal price from June 16, 2013 to June 25, 2013.

4.2.2. *Probabilistic forecasting of the electricity price*

We are now ready to forecast the electricity price distribution. We remarked that it was not necessary to consider non-linear effects here. Therefore, we only estimated $\log(P_t)$ by fitting linear quantile regression (see Section 2.2) without performing a first step to estimate non-linear effects. We used the following vector of covariates:

$$\begin{aligned}\boldsymbol{Z}_t = \big(&\log(P_t^{(\text{last})}),\ \log(P_t^{(\max)}),\ \log(P_{t-24}),\ \log(P_{t-48}), \\ &\log(P_{t-168}),\ \log(P_{t-24}^{(\min)}),\ DayType_t, \\ &FZL_t^{(0.95)},\ FTL_t^{(0.95)},\ FZL_t^{(0.8)},\ FTL_t^{(0.8)}\big)\,.\end{aligned}$$

All the covariates in $\boldsymbol{Z}_t$ are available 24 hours in advance except $\log(P_t^{(\max)})$. Thus, to address this problem, we average the forecasted distribution of the price over the forecasted distribution of the maximal price performed in Section 4.2.1. The method is similar to the one used for the temperature forecasts at the end of Section 3.2. Figure 10 displays the forecasted distribution and the observed prices for several days.

4.2.3. *Results*

We present in Table 2 and in Figure 11 the practical performance obtained by the method described in this section, denoted quantGAM, on the days evaluated by the competition. Table 2 also summarizes the results obtained by the team Tololo during the competition and by the two other methodologies quantMixt and quantGLM that will be detailed in the next sections. We note that quantGAM is especially robust to price

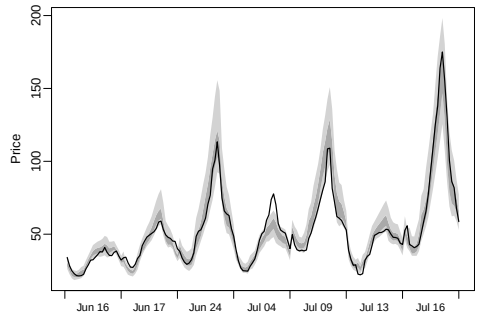

Figure 10: One day ahead forecasted distribution of the electricity price for days corresponding to tasks 1 to 7.

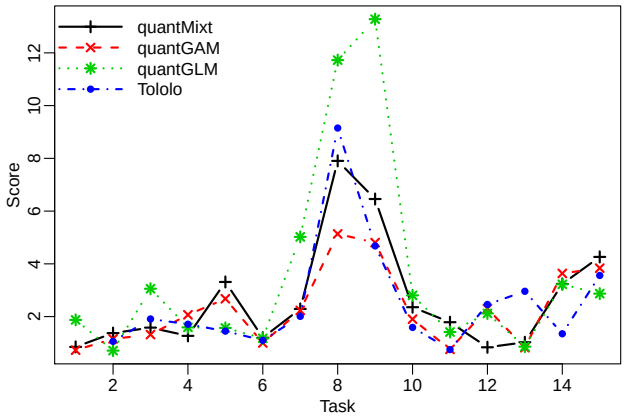

Figure 11: Performance of the different methods described in this article on the Tasks of the electricity price competition.

spikes (which occur on July 18 and July 19, 2013) and exhibits good performance over all tested days.

During the competition, we started by using from Task 2 to Task 8 several versions of quantMixt, then we used from Task 9 to Task 12 versions of quantGAM. For the last three tasks, that correspond to days in winter, we adopted quantGLM that is especially designed for winter. The results obtained by Tololo and by the three methods are different. This is mostly due to the fact that during the competition we did not use the same versions of the methods described in this paper. Indeed we constantly changed our methodology during the competition. Due to space constraint we cannot get into the details in this paper.

Table 2: Performance of quantGAM (Section 4.2), quantMixt (Section 4.3), and quantGLM (Section 4.4) together with the performance obtained by the benchmark (B.) and by the team Tololo (T.) on the tested days in 2013 of the electricity price competition. For each task the best of the results is highlighted in bold. The average performance over all days (other than June 06) is also reported.

Task	Date	B.	T.	quantGAM	quantMixt	quantGLM
1	Jun. 06	3.13	XX	**0.72**	0.85	1.87
2	Jun. 17	0.68	1.06	1.15	1.37	**0.71**
3	Jun. 24	8.13	1.91	**1.31**	1.58	3.05
4	Jul. 04	4.03	1.71	2.06	**1.27**	1.59
5	Jul. 09	7.97	1.45	2.67	3.31	**1.57**
6	Jul. 13	5.70	1.10	**0.99**	1.20	1.18
7	Jul. 16	12.15	2.01	**2.23**	2.28	5.02
8	Jul. 18	38.35	9.15	**5.13**	7.90	11.72
9	Jul. 19	44.23	4.68	**4.80**	6.45	13.27
10	Jul. 20	18.22	1.59	**1.90**	2.35	2.80
11	Jul. 24	31.57	0.75	**0.75**	1.78	1.42
12	Jul. 25	42.95	2.46	2.30	**0.84**	2.12
13	Dec. 06	2.86	2.96	**0.82**	1.03	0.86
14	Dec. 07	3.20	1.35	3.63	3.23	**3.22**
15	Dec. 17	22.38	3.56	3.83	4.26	**2.87**
Global		16.36	2.55	**2.40**	2.78	3.67

In the two remaining sections, we present the other methodologies that we considered for probabilistic electricity price forecasting. Although their results seem to be worse than quantGAM in Table 2, we think that they can be largely improved in the future.

4.3. *Combining individual predictors*

We present in this section a second approach that we used during the competition to form probabilistic forecast of the electricity price. We consider a methodology inspired from Nowotarski and Weron (2014). The idea is a two steps approach: first, we build a set of individual predictors that aims at predicting the mean of the electricity price; second, we combine these individual predictors so as to obtain a forecast of the quantiles by combining the individual predictors.

The training set is partitioned into two pieces. A *mixing set* that consists of the last 30 days of the training set (i.e., the last $30 \times 24 = 720$ hours) is used to learn the best combination of individual forecasters. The rest of the training set is the *fitting set* (data older than one month) which serves to fit the individual models.

4.3.1. *Individual Predictors*

We consider 13 individual predictors that were chosen because they exhibit various behaviors with the idea that the combining algorithm will be able to catch the best of each. We use the notations defined in Section 4.1.

A first class of individual forecasters are well-known predictors that have been proven to perform well in electricity price forecasting. They include several autoregressive models, spike pre-processed autoregressive models, and threshold autoregressive models described more in details in Nowotarski and Weron (2014):

1. An autoregressive model (`AR`) defined as:

$$\log(P_t) = \alpha_1 \log(P_{t-24}) + \alpha_2 \log(P_{t-48}) + \alpha_3 \log(P_{t-168}) + \alpha_4 \log(P_{t-24}^{(\min)}) + h(DayType_t) + \varepsilon_t \,,$$

where ε_t (as in the other models) are i.i.d. centered Gaussian noise, α_i are linear coefficient to be estimated, and h is a real function to be estimated as in Equation (10).

2. An autoregressive model with forecasted electric loads as additional covariates (`ARX`). It is defined as:

$$\begin{aligned}\log(P_t) = \; & \alpha_1 \log(P_{t-24}) + \alpha_2 \log(P_{t-48}) + \alpha_3 \log(P_{t-168}) \\ & + \alpha_4 \log(P_{t-24}^{(\min)}) + \alpha_5 \log(FTL_t) \\ & + \alpha_6 \log(FZL_t) + h(DayType_t) + \varepsilon_t \,.\end{aligned}$$

3. A threshold autoregressive model `TAR` defined as an extension of `AR` to two regimes depending on the variation of the mean price between a day and eight days ago.

4. `TARX` the extension of `ARX` to the two regimes model.

5. Spike pre-processed autoregressive model (`PAR`): the idea is to pre-process the price data by removing large spike (see the "damping scheme" presented in Weron and Misiorek 2008): for all $P_t > M$ set $\tilde{P}_t = M + M \log_{10}(P_t/M)$ where M is a fixed parameter set to the mean price plus three standard deviations. Then `AR` is fitted by substituting the prices $P_t, P_{t-24}, \ldots$ with the pre-processed prices $\tilde{P}_t, \tilde{P}_{t-24}, \ldots$.

6. `PARX` similar to `PAR`, but `ARX` is fitted with pre-processed prices.

The seven remaining individual forecasters are designed by considering several regression methods, several subsets of covariates, and different sets of fitting data. More precisely we list them hereafter:

7. A linear regression (function `lm` in `R`) fitted with model:

$$\begin{aligned}\log(P_t) = \; & \alpha_1 \log(P_{t-24}) + \alpha_2 \log(P_{t-48}) + \alpha_3 \log(P_{t-168}) \\ & + \alpha_4 \log(P_t^{(\max)}) + \alpha_5 FZL_t^{(0.95)} + \alpha_6 FTL_t^{(0.95)} \\ & + \alpha_7 FZL_t^{(0.8)} + \alpha_8 FTL_t^{(0.8)} + h(DayType_t) + \varepsilon_t\end{aligned}$$

   In order to produce a forecast, $P_t^{(\max)}$ is substituted with its forecast performed by a generalized additive model with Equation (15). This substitution is also performed for the next individual predictors.

8. A linear regression (`lm`) fitted as follows:

$$\begin{aligned}\log(P_t) = \; & \alpha_1 \log(P_t^{(\text{last})}) + \alpha_2 \log(P_{t-24}) + \alpha_3 \log(P_t^{(\max)}) \\ & + \alpha_4 FTL_t + \alpha_5 FZL_t + h(DayType_t) + \varepsilon_t \,.\end{aligned}$$

9. A generalized additive model (see Section 2.1 for details) fitted with:

$$\begin{aligned}\log(P_t) = \; & f_1(ToY_t) + f_2(\log(P_{t-24})) + f_3(\log(P_t^{(\max)})) \\ & + f_4(\log(P_{t-24}^{(\text{mean})})) + h(DayType_t) + \varepsilon_t \,.\end{aligned}$$

   We recall that f_1 is a cubic cyclic regression spline, f_2, f_3, and f_4 are cubic regression splines, and h is some real function as in Equation (10).

10. A generalized additive model fitted with:

$$\begin{aligned}\log(P_t) = \; & f_1(\log(P_{t-24})) + f_2(\log(P_t^{(\max)})) \\ & + f_3(\log(P_{t-24}^{(\text{mean})})) + f_4(FTL_t) + f_5(FZL_t) \\ & + f_6(ToY_t) + h(DayType_t) + \varepsilon_t \,.\end{aligned}$$

    This model was only trained on Summer data (i.e., from June to August) if the day to be predicted is in Summer itself. The idea of creating specialized forecasters in order to obtain more diversity was investigated in Gaillard and Goude (2014) in the context of electricity consumption forecasting.

Then we considered two forecasters using random forests regression. Random forests were introduced by Breiman (2001) and are available in the `R`-package `randomforest`[3]. They are a powerful ensemble method that builds a large number of random regression binary trees before aggregating them, so as to improve their prediction accuracy. The process of fitting a random forests model and using it to produce a prediction performs three steps. First it performs bagging: it creates multiple bootstrap samples of the training set by sampling from the training set n observations uniformly and with replacement, where n is the size of the training set. Second it builds random prediction trees: for each bootstrap sample, it builds a corresponding binary decision tree by partitioning the covariate space recursively in a dyadic fashion. The trees are extended very deeply in practice and have high variance and low bias. Third it forms a prediction, by averaging the predictions of the individual regression trees. The two random forests individual predictors are:

11. Random forests regression fitted with all covariates described in the previous models.

12. Random forests regression fitted with the same covariates used by the individual predictor 8.

Our last predictor is trained by using gradient boosted methods. Gradient boosted methods were introduced by Friedman (1999) and are implemented in the `R`-package `gbm`. It is another ensemble method. By contrast to random forests the basis forecasters are a class of weak regression methods (e.g., decision trees with very limited maximal depth) and are built sequentially by trying to reduce the bias of the combined predictor. We ran the experiments by using the default parameters of the package `gbm` and by optimizing the number of trees (basis forecasters) on the out-of-bag error using the function `gbm.perf`. The last predictor is:

13. Gradient boosting machine fitted with the same covariates used by the individual predictor 8.

In the end we have 13 individual forecasters of the price. The exact form and number of individual forecasters changed

[3] we run all experiments with default parameters of the package

over time and over our submissions. For the sake of simplicity, we only presented in this paper few predictors among those we tested. This partly explains why the results obtained by the methods presented in this paper are different from those obtained during the competition.

4.3.2. Combining forecasts

Once the forecasters have been designed and fitted on the *fitting set*, we learn on the *mixing set* (the last thirty days of available data, that we denote by E) how to combine them in order to provide probabilistic forecasts. To do so, we consider the setting of online robust aggregation of predictors (see the monograph of Cesa-Bianchi and Lugosi (2006) for a nice overview) which was sucessfully applied on electricity load data in see Devaine et al. (2013).

We consider a version of the ML-Poly forecaster introduced in Gaillard et al. (2014) because it is fully adaptive and was proven to exhibit good performance on the electric load signal (see Gaillard and Goude 2014). We detail it now. Let us denote for each time t by $\boldsymbol{x}_t = (x_{t,1}, \dots, x_{t,K}) \in \mathbb{R}_+^K$ the forecasts of P_t produced by the $K = 13$ individual forecasters described in Section 4.3.1. Our algorithm is described as Algorithm 1. For each quantile $\tau \in (0,1)$, and for $\beta > 0$, it consists in performing a kind of stochastic gradient descent so as to minimize in $\boldsymbol{\theta} \in \mathcal{B}_1(\beta) \triangleq \{\boldsymbol{\theta} \in \mathbb{R}^K, \|\boldsymbol{\theta}\|_1 \leq \beta\}$ a regularized version of the average pinball loss over the mixing set: $1/(\mathrm{Card}\, E) \sum_{t \in E} \rho_\tau(P_t - \boldsymbol{\theta}^\top \boldsymbol{x}_t)$. Here, Card E denotes the number of observations in the *mixing set* (i.e., $24 \times 30 = 720$ hours), and ρ_τ is the pinball loss defined in Section (2.2).

More precisely, we set $m \in \mathbb{N}^*$ a number of optimization steps. At each instance $i \in \{1, \dots, m\}$, Algorithm 1 samples an observation t_i uniformly in the *mixing set* E (Step 1). Then it updates (Step 3) a weight-vector $\widehat{\boldsymbol{p}}_i$ to $\widehat{\boldsymbol{p}}_{i+1}$ in the simplex $\Delta_{2K} \triangleq \{x \in \mathbb{R}_+^{2K} \ : \ \sum_k x_k = 1\}$ in order to improve the τ-quantile prediction of P_{t_i} by the weighted average $\widehat{P}_i = \sum_{k=1}^{2K} \widehat{p}_{i,k} \tilde{x}_{t_i,k}$, where for all times $t \geq 1$,

$$\tilde{\boldsymbol{x}}_t \triangleq \beta \left(-x_{t,1},\, x_{t,1}, \dots, -x_{t,K},\, x_{t,K} \right).$$

Doing so, it can be guaranteed (see Gaillard et al. 2014; Cesa-Bianchi and Lugosi 2006) under i.i.d. assumption of the data that the risk of the average weight vector $\bar{\boldsymbol{p}}_\tau = (1/m) \sum_{i=1}^m \widehat{\boldsymbol{p}}_i$ is close to the optimal risk $\min_{p \in \Delta_{2K}} \mathbb{E}[\rho_\tau(P_t - \boldsymbol{p}^\top \tilde{x}_t) | x_t]$. The returned combination vector is finally $\widehat{\boldsymbol{\theta}}_\tau$ defined in (16) because $\widehat{\boldsymbol{\theta}}_\tau^\top \boldsymbol{x}_t = \bar{\boldsymbol{p}}_\tau^\top \tilde{\boldsymbol{x}}_t$. The idea of considering convex combinations of $\tilde{\boldsymbol{x}}_t$ in Δ_{2K} in order to perform combinations of $\boldsymbol{x}_t$ in the linear ball $\mathcal{B}_1(\beta)$ dates back to Kivinen and Warmuth (1997).

In order to produce a forecast of the τ-quantile of P_t, we run over the *mixing set* E Algorithm 1 with parameters $m = 5000$ and $\beta = 2$ and we predict $\widehat{q}_{t,\tau} = \sum_{k=1}^K \widehat{\theta}_{\tau,k} x_{t,k}$, where $\boldsymbol{x}_t$ are the forecasts of the individual forecasters defined in Section 4.3.1. We call quantMixt this method.

Figure 12 plots the forecasted price distribution together with the observed price values and the individual forecasts for June 16 and June 17, 2013. The performance of this method is reported in Table 2. The performance is good except on July 18 and July 19, 2013 that correspond to spikes in the electricity price. We remarked that this was partly due to bad predictive performance of individual forecasts which did not detect spikes and partly due to the relative small size of the *mixing set* where few electricity spikes are observed. This suggests two possible directions for future research to improve the method. The first avenue would be to improve the set of individual predictors. The second one would be to consider sequential versions of the individual predictors in order to learn the combinations over the whole training set.

Input: $\tau \in (0,1),\ \beta > 0,\ m \in \mathbb{N}^*$
Initialize: $\widehat{\boldsymbol{p}}_1 = (1/(2K), \dots, 1/(2K)) \in \Delta_{2K}$

for each step $i = 1, 2 \dots, m$ **do**

1. sample t_i uniformly in E

2. define for all $k = \{1, \dots, 2K\}$

$$\tilde{x}_{t_i,k} = \left(\mathbb{I}_{\{k \text{ is even}\}} - \mathbb{I}_{\{k \text{ is odd}\}} \right) \beta\, x_{t_i, \lceil k/2 \rceil}$$

and the learning rate

$$\eta_{i,k} = \left(1 + \sum_{s=1}^i \left(\ell_s(\widehat{P}_s) - \ell_s(\tilde{x}_{t_s,k}) \right)^2 \right)^{-1}$$

where $\ell_s : x \mapsto \left(\mathbb{I}_{\{P_{t_s} < \widehat{P}_s\}} - \tau \right) x$ and $\widehat{P}_s = \widehat{\boldsymbol{p}}_s^\top \tilde{\boldsymbol{x}}_{t_s}$.

3. form the mixture $\widehat{\boldsymbol{p}}_{i+1} \in \Delta_{2K}$ component-wise by

$$\widehat{p}_{i+1,k} = \frac{\eta_{s,k} \left(\sum_{s=1}^i \ell_s(\widehat{P}_s) - \ell_s(\tilde{x}_{t_s,k}) \right)_+}{\sum_{j=1}^{2K} \eta_{s,j} \left(\sum_{s=1}^i \ell_s(\widehat{P}_s) - \ell_s(\tilde{x}_{t_s,j}) \right)_+} \in [0,1]$$

where $x_+ \triangleq \max\{x, 0\}$.

end for

Return the weight vector $\widehat{\boldsymbol{\theta}}_\tau \in [-\beta, \beta]^K$ component-wise defined as

$$\widehat{\theta}_{\tau,k} = \tfrac{1}{m} \sum_{i=1}^m \beta \left(p_{i,2k} - p_{i,2k+1} \right) \tag{16}$$

Algorithm 1: The averaged linear ML-Poly weighted average forecaster for quantile prediction

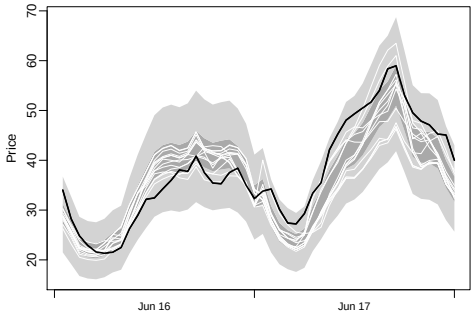

Figure 12: One day ahead forecasted distribution of the electricity price for days corresponding to tasks 1 and 2. The black line depicts the observed electricity price and the white lines plots the individual forecasts.

4.4. Kernel based quantile regression with Lasso penalty

Here we present the last method that we implemented in this competition for electricity price forecasting. It was motivated by the fact that after twelve weeks of competition we have generated a lot of covariates from the three original ones: total and zonal loads and prices, and we want to take advantage of that in an automatic fashion. Here is the list of those transformations, where X_t could be either the price P_t, the zonal load FZL_t or the total load FTL_t at time t:

- lagging: we consider the following lagged variables X_{t-24}, X_{t-48}, X_{t-168}, X_{t-336}, and $X_t^{(\text{last})}$ the last available observation at the time of the forecast;
- log and log log transforms: $\log(x_t)$, $\log\big(\log(x_t)\big)$;
- spike preprocessing of the price (see models TARX and ARX of Section 4.3.1).
- mean, max and min of the day corresponding to observation t: $X_t^{(\min)}$, $X_t^{(\max)}$, and $X_t^{(\text{mean})}$ (see Section 4.1);
- exponential smoothing with parameters $\gamma \in \{0.8, 0.95\}$ defined by induction as in Equation (7).

All these single transformations could be coupled: for example we could take the log of a lagged covariates or the maximum of a smoothed covariate etc. In the end by adding also the calendar variables $DayType_t$ we obtain $d \triangleq 192$ covariates that are candidates to enter into a linear quantile regression. These covariates are potentially highly correlated. In the following we denote by $\boldsymbol{X}_t \in \mathbb{R}^d$ the vector that contains all transformations of covariates. Our idea is to select among those 192 covariates the ones which produce good forecasting performance for each quantile. To this end we propose to use a ℓ_1 selection procedure as presented in Tibshirani (1996) together with a kernel regularized regression. This approach has already been studied theoretically in Belloni and Chernozhukov (2011) but we did not find any existing `R` package. To benefit from the nice performance of the `R` package `glmnet` (see Friedman et al. 2010) we propose a two steps approach detailed hereafter. First we fit a single ℓ_1-regression model on the mean by using `glmnet` with the Lasso penalty (cf. Step 1). Then we fit a quantile regression model on the residuals of this model by using a weighted version of the previous algorithm with weights corresponding to a Gaussian kernel centered around each quantile (cf. Step 2).

- Step 1: we estimate the mean price by a sparse linear combination of the covariates. To do so, we solve the optimization problem:

$$\widehat{\boldsymbol{\beta}} \in \underset{\boldsymbol{\beta}\in\mathbb{R}^d}{\arg\min} \left\{ \sum_{t=1}^{n} (P_t - \boldsymbol{X}_t^\top \boldsymbol{\beta})^2 + \lambda \|\boldsymbol{\beta}\|_1 \right\}$$

  where $d = 192$ denotes the number of covariates, n is the number of observations in the training set, $\boldsymbol{X}_t \in \mathbb{R}^d$ is the vector of covariates, and $\lambda > 0$ is a parameter that penalizes large models. It has to be optimized. We obtain the estimate $\widehat{\boldsymbol{\beta}} \in \mathbb{R}^d$ and the residual signal $\widehat{\varepsilon_t} \triangleq P_t - \boldsymbol{X}_t^\top \widehat{\boldsymbol{\beta}}$.

- Step 2: for each quantile $\tau \in (0, 1)$, we estimate a correction to add to the mean estimates $\boldsymbol{X}_t^\top \widehat{\boldsymbol{\beta}}$ in order to estimate the quantile of level τ. To achieve this, we first compute e_τ the empirical quantile of the sequence $(\widehat{\varepsilon_t})_{t=1,\dots,n}$ and the weights for all times $t = 1, \dots, n$:

$$w_{\tau,t} = \frac{\exp\big(-(\widehat{\varepsilon_t} - e_\tau)^2/h\big)}{\sum_{t=1}^{n} \exp\big(-(\widehat{\varepsilon_t} - e_\tau)^2/h\big)}$$

  where $h > 0$ is a window parameter to be optimized. Then we proceed to the optimization problem:

$$\widehat{\boldsymbol{\beta}}_\tau \in \underset{\boldsymbol{\beta}_\tau\in\mathbb{R}^d}{\arg\min} \left\{ \sum_{t=1}^{n} w_{\tau,t} \left(\widehat{\varepsilon_t} - \boldsymbol{X}_t^\top \boldsymbol{\beta}_\tau\right)^2 + \lambda \left\|\boldsymbol{\beta}_\tau\right\|_1 \right\}.$$

- Step 3: the final quantile forecast $\widehat{q}_{\tau,t}$ of the price at time t is then obtained by

$$\widehat{q}_{\tau,t} \triangleq \underbrace{\boldsymbol{X}_t^\top \widehat{\boldsymbol{\beta}}}_{\text{mean estimate}} + \underbrace{\boldsymbol{X}_t^\top \widehat{\boldsymbol{\beta}}_\tau}_{\tau\text{-quantile correction}}.$$

In Steps 1 and 2, the optimal penalization parameter λ is found by using 10-fold cross-validation implemented in the `glmnet` package. The window parameter h has been optimized on a grid over the last winter period (because we used this method at the end of the competition in order to predict days in winter). As the process is time consuming, Step 2 is actually only performed for quantiles $\tau \in \{0.01, 0.99\} \cup \{0.1, 0.2, \dots, 0.9\}$. The other quantiles are formed by linear interpolation between those quantiles.

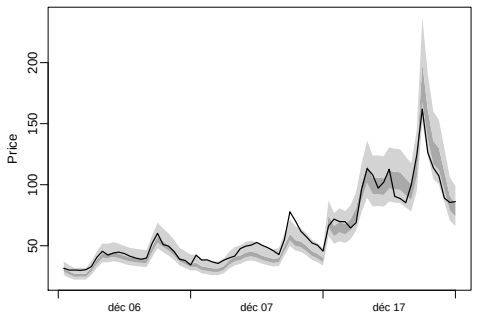

Figure 13: One day ahead forecasted distribution by quantGLM of the electricity price for days corresponding to tasks 13, 14, and 15.

The performance of this method (denoted quantGLM) is reported in Table 2 and quantile forecasts are represented on Figure 13. Note that the parameter of the method were optimized on winter which may explains its bad performance of several days in summer. Another critical point is that the covariate selection is mostly done at Step 1 inducing potential bias for some quantile that could not be corrected at Step 2. We leave for future research a unified algorithm for covariate selection for each quantile avoiding these kind of bias as well as the optimization of the method in summer.

4.5. Perspectives

In the end, we proposed for the probabilistic price forecasting task three methods (quantGAM, quantMixt, and quantGLM)

that achieve good performance (cf. Table 2). We think that they can be largely improved. Figure 14 plots the percentage of time the observed electricity price P_t is smaller than the quantiles $\widehat{q}_{\tau,t}$ predicted by these methods according to $\tau \in (0,1)$. The closer the curve is from the identity function the better. While quantGAM and quantMixt do not seem to be biased, quantGLM used to overestimate low quantiles and underestimate high quantiles. Understanding this behavior may yield a possible improvement of quantGLM in the future.

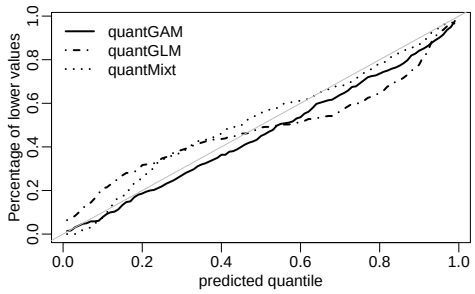

Figure 14: Percentage of observed electric load values P_t under the predicted quantiles $\widehat{q}_{\tau,t}$.

5. Conclusion

We propose a new methodology for probabilistic forecasting based on GAMs and quantile regression that achieves good results (rank 1^{st}) in the load and price forecasting tracks. For the price forecasting track we compare it to two other competitive approaches, combining individual predictors and kernel based quantile regression with lasso penalty for covariate selection. In the end, quantGAM is easy to implement, computationally fast and performed better on these datasets. Furthermore, it offers a good interpretation of the effects that impact the electricity demand or price. There is still some work to assess theoretical properties of quantGAM.

Acknowledgements. We thank the competition organizers for organizing this contest which was a rewarding experience. We would also like to thank Ghislain Agoua for insightful initial work on quantile GAM carried out during his internship at EDF R&D.

Belloni, A., Chernozhukov, V., 02 2011. l1-penalized quantile regression in high-dimensional sparse models. Ann. Statist. 39 (1), 82–130.

Breiman, L., Oct. 2001. Random forests. Mach. Learn. 45 (1), 5–32.

Cesa-Bianchi, N., Lugosi, G., 2006. Prediction, learning, and games. Cambridge University Press.

Devaine, M., Gaillard, P., Goude, Y., Stoltz, G., 2013. Forecasting electricity consumption by aggregating specialized experts – a review of the sequential aggregation of specialized experts, with an application to Slovakian and French country-wide one-day-ahead (half-)hourly predictions. Machine Learning 90 (2), 231–260.

Friedman, 1999. Stochastic gradient boosting. Computational Statistics and Data Analysis 38, 367–378.

Friedman, J. H., Hastie, T., Tibshirani, R., 2 2010. Regularization paths for generalized linear models via coordinate descent. Journal of Statistical Software 33 (1), 1–22.

Gaillard, P., Goude, Y., 2014. Forecasting the electricity consumption by aggregating experts; how to design a good set of experts. In: Antoniadis, A., Brossat, X., Poggi, J.-M. (Eds.), Modeling and Stochastic Learning for Forecasting in High Dimension. Springer, to appear.

Gaillard, P., Stoltz, G., van Erven, T., 2014. A second-order bound with excess losses. In: Proceedings of COLT.

Goude, Y., Nedellec, R., Kong, N., 2012. Local short and middle term electricity load forecasting with semi-parametric additive models. submitted to IEEE transactions on smart grid.

Hastie, T., Tibshirani, R., 1990. Generalized Additive Models. Chapman & Hall/CRC.

Hong, T., Pinson, P., Fan, S., Zareipour, H., Troccoli, A., Hyndman, R., 2015. Probabilistic Energy Forecasting: State-of-the-art 2015, international Journal of Forecasting. To appear.

Kivinen, J., Warmuth, M. K., 1997. Exponentiated gradient versus gradient descent for linear predictors. Information and Computation 132 (1), 1 – 63.

Koenker, R., 2013. quantreg: Quantile Regression. R package version 5.05. URL `http://CRAN.R-project.org/package=quantreg`

Koenker, R. W., Bassett, G. W., 1978. Regression quantiles. Econometrica 46 (1), 33–50.

Nowotarski, J., Weron, R., 2014. Computing electricity spot price prediction intervals using quantile regression and forecast averaging. Computational Statistics, 1–13.

Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. J. Roy. Statist. Soc. Ser. B 58 (1), 267–288.

Weron, R., Misiorek, A., 2008. Forecasting spot electricity prices: A comparison of parametric and semiparametric time series models. International Journal of Forecasting 24 (4), 744 – 763.

Wood, S., 2006. Generalized Additive Models, An Introduction with R. Chapman and Hall.