Forecasting Electricity Consumption by Aggregating Experts; How to Design a Good Set of Experts

Pierre Gaillard and Yannig Goude

Abstract Short-term electricity forecasting has been studied for years at EDF and different forecasting models were developed from various fields of statistics or machine learning (functional data analysis, time series, non-parametric regression, boosting, bagging). We are interested in the forecasting of France's daily electricity load consumption based on these different approaches. We investigate in this empirical study how to use them to improve prediction accuracy. First, we show how combining members of the original set of forecasts can lead to a significant improvement. Second, we explore how to build various and heterogeneous forecasts from these models and analyze how we can aggregate them to get even better predictions.

1 Introduction

Electricity consumption forecasting is a crucial matter for electricity providers like EDF to maintain the equilibrium between production and demand. Overestimating the consumption leads to overproduction, which has a negative environmental impact and implies unnecessary loss of benefits for the company. On the other hand, underestimating the consumption may cause a shortage of energy and black outs. In the past years EDF R&D has therefore developed several competitive forecasting models achieving around 1.4 % error in MAPE (the average of percentage errors, see (2) for a formal definition) at the daily horizon. However the electrical scene in France is constantly evolving (nuclear power, electric cars, air conditioning are developing for instance) and the opening of the electricity market induces potential

P. Gaillard (✉)
EDF R&D, 1 av du Général de Gaulle, Clamart, France

GREGHEC, CNRS, Jouy-en-Josas, France
e-mail: pierre@gaillard.me

Y. Goude
EDF R&D, 1 av du Général de Gaulle, Clamart, France
e-mail: yannig.goude@edf.fr

A. Antoniadis et al. (eds.), *Modeling and Stochastic Learning for Forecasting in High Dimensions*, Lecture Notes in Statistics 217,
DOI 10.1007/978-3-319-18732-7_6

customer losses. Therefore the historical models have to be regularly reconsidered and challenged. As daily forecasts are the main inputs for optimizing the production units we consider in this paper the goal of improving short-term (daily) forecasting of France's electricity consumption.

As the historical French electricity provider, EDF has investigated the issue of load forecasting for years and developed models from a wide range of statistical or machine learning methods. Among many, we consider in this study three approaches presented below. They were chosen for two main reasons. First, they have a good forecasting accuracy. Second, they are derived from quite different statistical frameworks, which results in a sort of heterogeneity. The first model is a non-parametric model based on regularized regression on spline basis (see Wood [28]). It will be referred to next as the generalized additive model (GAM). This model has performed well on France's load consumption signal (see Pierrot and Goude [25]), on EDF portfolio data (see Wood et al. [29]) and was proven to be a good competitor on US data (see Nedellec et al. [24]). The second model is based on curve linear regression (CLR) via dimension reduction. It is introduced and applied to electricity consumption forecasting in Cho et al. [10, 11]. The third and last model, kernel wavelet functional (KWF), is detailed in Antoniadis et al. [2–4]. It combines clustering functional data and detection of similar patterns in functional processes based on a wavelet distance. These three approaches are based on extremely different insights and we expect it can induce different behaviors that an aggregation algorithm can take advantage of in some online fashion. The GAM model captures non-linear relationships between electricity load and the different covariates driving it (temperature, fare effects...) and provides smooth estimates of these transfer functions without any transformation of the original data. The CLR model performs a data-driven dimension reduction as well as a data transformation so that the relationship between the transformed data is linear and can be captured by simple multivariate regression models. The KWF approach is non-parametric and does not use any exogenous variable but the past consumption. It is particularly robust to special days (bank holidays, holiday seasons) and meteorological forecasts errors. In the GAM setting, observations (half-hourly electricity load and covariates) are considered as finite dimensional whereas in the CLR and the KWF approaches, daily electricity load is the realization of a functional process.

As we have at our disposal three forecasting models, a straightforward question is how to combine them to produce accurate forecasts. The art of combining forecasts has been extensively studied for the past four decades (see the review of Clemen [12]) and the empirical literature is voluminous. However, few real-world empirical studies consider the framework of individual sequences to design the aggregation rules. Some of them include for instance climate prediction [23], air-quality prediction [21, 22], quantile prediction of daily call volumes entering call center [6], or electricity consumption [13]. The vast majority of these studies focuses however on the aggregation rules and how to weight the experts. Little consideration goes into designing the set of experts to include in the combination. Aiolfi et al. in their technical report [1] studied the construction of a varied enough set of experts by considering the combination of linear autoregressive models with

non-linear models (logistic smooth transition autoregressive and neural networks). They however did not consider the same aggregation rules as we do: because of the small length of their time series, none of their rules had time to learn the weights and the best results were obtained using uniform aggregation scheme.

We now describe the methodology followed in this study. We aim first at designing a set of base forecasting methods (henceforth referred to as experts) by using the three models described above. We show how an aggregation rule that sequentially outputs forecasts of the electricity consumption for the next instances can significantly improve upon these experts. The aggregation rules and the framework of prediction with expert advice is detailed in Sect. 2. Then, we propose different strategies to design a larger set of experts from the three initial experts and give a detailed analysis of the corresponding combined forecasts.

2 Sequential Aggregation of Experts

The content of this section reviews the framework of sequential prediction with expert advice, a setting which received considerable attention in the past 20 years (see the monograph by Cesa-Bianchi and Lugosi [9]). It considers an online learning scenario in which a forecaster has to guess element by element future values of an observed time series. To form its prediction it receives and combines before each instance the opinions of a finite set of experts. This framework makes possible to consider several stochastic models with extremely different assumptions in a single approach. To do so, it adopts the deterministic and robust point of view of the literature of individual sequences. It is thus particularly adapted to our application.

2.1 Mathematical Context

We now present the mathematical setting of prediction with expert advice. We suppose that at each time instance $t = 1, \ldots, T$ the next outcome y_t of a sequence of observations $y_1, \ldots, y_T$, like half-hourly electricity consumptions, is to be predicted. We assume that the observations are all bounded by some positive constant B, so that $y_t \in [0, B]$. Before each time instance t, a finite number K of experts provide forecasts $\boldsymbol{x}_t = (x_{1,t}, \ldots, x_{K,t}) \in [0, B]^K$ of the next observation y_t. A forecaster is then asked to form its own prediction with knowledge of the past observations $y_1^{t-1} = y_1, \ldots, y_{t-1}$ and of the past expert advice $\boldsymbol{x}_1^t = \boldsymbol{x}_1, \ldots, \boldsymbol{x}_t$. Let denote by $\cdot$ the inner product in $\mathbb{R}^K$. Formally the forecaster forms a mixture $\hat{\boldsymbol{p}}_t = (\hat{p}_{1,t}, \ldots, \hat{p}_{K,t}) \in \mathbb{R}^K$ and predicts $\hat{y}_t = \hat{\boldsymbol{p}}_t \cdot \boldsymbol{x}_t = \sum_{k=1}^K p_{k,t} x_{k,t}$ by linearly combining the predictions of the experts.

The accuracy of a prediction x proposed by an expert or by the aggregation rule at time instance t for the outcome y_t is measured through a convex loss function ℓ_t. In this paper, we consider the special case of the square loss $\ell_t(x) = (y_t - x)^2$. The

analysis can however be easily extended to any convex loss function. On instance t, expert k suffers loss $\ell_t(x_{k,t}) = (y_t - x_{k,t})^2$ and the aggregation rule incurs loss $\ell_t(\hat{y}_t) = (y_t - \hat{y}_t)^2$. The goal of the forecaster is to design aggregation rules (that is, applications $\mathscr{A} : (\boldsymbol{x}_1^t, y_1^{t-1}) \mapsto \hat{\boldsymbol{p}}_t$) with small average error. The latter can be decomposed as

$$\frac{1}{T}\sum_{t=1}^{T}(y_t - \hat{y}_t)^2 \quad \triangleq \quad \inf_{\boldsymbol{q}\in S}\left\{\frac{1}{T}\sum_{t=1}^{T}(y_t - \boldsymbol{q}\cdot\boldsymbol{x}_t)^2\right\} + R_T\,, \tag{1}$$

where S is some closed and bounded subset of $\mathbb{R}^K$; and this defines the regret R_T. As we explain next this decomposition highlights the well-known trade-off between approximation error and estimation error. Because these two terms add up to the error incurred by the aggregation rule they act as two opposing forces.

The first term in (1) is the error encountered by the best constant weight vector chosen in hindsight in a closed and bounded set $S \subset \mathbb{R}^K$. This best mixture is called an *oracle*. Its performance is the target that the aggregation rule intends to reach and is thus used as a benchmark value to be compared to the performance of an aggregation rule. Several oracles can be defined according to the set S the aggregation rule aims at competing with. We can list several oracles: the *best expert* oracle suffers $\min_{k=1,\dots,K}\sum_{t=1}^{T}(y_t - x_{k,t})^2$; the *best convex weight vector* corresponds to the best element in $S = \Delta_K \triangleq \{\boldsymbol{q} \in \mathbb{R}_+^K \,:\, \sum_i q_i = 1\}$; and finally the *best linear* oracle is defined by $S = B_K(r)$ the ball of radius r in $\mathbb{R}^K$. The larger the set S we aim at competing with, the smaller the first term in (1) is, but the harder it is for the aggregation rule to remain competitive. The second term grows in general. This approximation error is closely related to the expert forecasts. It decreases with increasing heterogeneity of the expert set.

The second term R_T is the estimation error. It evaluates the ability of the aggregation rule to retrieve online the oracle, i.e., the best possible mixture. If the aggregation rule is well designed, R_T will vanish to 0 as the length T of the experiment grows to infinity.

We assume in this paper that we have an efficient aggregation rule and we focus on reducing the approximation error; indeed many efficient aggregation rules are already well-known—see Sect. 2.2, but the approximation error is often left out of the debate.

2.2 Aggregation Rules

Experiments are performed by considering four different aggregation rules: the exponentially weighted average forecaster (EWA), the fixed share forecaster (FS), the ridge regression forecaster (Ridge), and the polynomially weighted average forecaster with multiple learning rates (ML-Poly). EWA, FS, and Ridge are described in the book of Cesa-Bianchi and Lugosi [9] for constant values of their

learning parameters. Devaine et al. [13] already applied EWA and FS to short-term load forecasting. They suggested in Sect. 2.4 an empirical tuning of the learning parameters which comes with no theoretical guarantees but works empirically well. It consists of optimally choosing the learning parameters on adaptive finite grids. Except for ML-Poly which already comes with its own learning parameter calibration rule, the parameters are tuned online following the method of Devaine et al. [13].

The exponentially weighted average forecaster (EWA) is an online convex aggregation rule introduced in learning theory by Littlestone and Warmuth [20] and by Vovk [27]. At time instance t, it assigns to expert k the weight

$$\hat{p}_{k,t} = \frac{e^{-\eta \sum_{s=1}^{t-1} \ell_s(x_{k,s})}}{\sum_{i=1}^{K} e^{-\eta \sum_{s=1}^{t-1} \ell_s(x_{i,s})}},$$

which is exponentially small in the cumulative loss suffered so far by the expert. When the learning parameter η is properly tuned, it has a small average regret $R_T = O\left(1/\sqrt{T}\right)$ with respect to the best fixed expert oracle—see Cesa-Bianchi and Lugosi [9].

The fixed share forecaster (FS) is due to Herbster and Warmuth [18]. It has the property to compete not only with the best fixed expert but with the best sequence of experts that may change a small number of times. It is particularly interesting when dealing with non stationary environments, in which the best expert should regularly be reconsidered. The fixed share forecaster considers a learning parameter η as well as a mixing parameter $\alpha \in [0, 1]$ that evaluates the number of changes in the oracle sequence of experts we are competing with.

We now provide a short mathematical description of the fixed share aggregation rule. The initial weight distribution is uniform $\hat{p}_1 = (1/K, \ldots, 1/K)$. Then, at each instance t, the weights are updated twice. First, a *loss update* takes into account the new loss incurred by each expert,

$$\hat{v}_{k,t} = \frac{\hat{p}_{k,t-1} e^{-\eta \sum_{s=1}^{t-1} \ell_s(x_{k,s})}}{\sum_{i=1}^{K} \hat{p}_{i,t-1} e^{-\eta \sum_{s=1}^{t-1} \ell_s(x_{i,s})}}.$$

Second a *mixing-update* ensures that each expert gets a minimal weight α/K by assigning

$$\hat{p}_{k,t} = (1 - \alpha)\hat{v}_{k,t} + \alpha/K.$$

This update captures the possibility that the best expert may have switched at time instance t. The fixed share forecaster was proven to have nice theoretical properties and vanishing average regret R_T with respect to sequences of experts with few shifts.

Algorithm 1: The polynomially weighted average forecaster with multiple learning rates (ML-Poly)

Initialization: $\boldsymbol{p}_1 = (1/K, \ldots, 1/K)$ and $\boldsymbol{R}_0 = (0, \ldots, 0)$
For each instance $t = 1, 2, \ldots, T$

0. pick the learning rates

$$\eta_{k,t-1} = 1/\left(1 + \sum_{s=1}^{t-1} \big(\ell_s(\hat{y}_s) - \ell_s(x_{k,s})\big)^2\right)$$

1. form the mixture $\hat{\boldsymbol{p}}_t$ defined component-wise by

$$\hat{p}_{k,t} = \eta_{k,t-1} \big(R_{k,t-1}\big)_+ \;/\; \boldsymbol{\eta}_{t-1} \cdot (\boldsymbol{R}_{t-1})_+$$

   where $\boldsymbol{x}_+$ denotes the vector of non-negative parts of the components of $\boldsymbol{x}$
2. output prediction $\hat{y}_t = \widehat{\boldsymbol{p}}_t \cdot \boldsymbol{x}_t$
3. for each expert k update the regret

$$R_{k,t} = R_{k,t-1} + \ell_t(\hat{y}_t) - \ell_t(x_{k,t})$$

For more details about the fixed share aggregation rule the reader is referred to Cesa-Bianchi and Lugosi [9, Section 5.2].

The polynomially weighted average forecaster with multiple learning rates (ML-Poly) is obtained via a version of the polynomially weighted average forecaster detailed in Cesa-Bianchi and Lugosi [8], see also Cesa-Bianchi and Lugosi [9, Section 2.1]. The multiple learning rate version is due to Gaillard et al. [17] whose implementation is recalled in Algorithm 1. Gaillard et al. [17] proved the regret bound $R_T = O\big(1/\sqrt{T}\big)$ with respect to the best fixed expert. ML-Poly is particularly interesting since despite the theoretical tuning of the learning parameters, it achieves as good performance as the other ones. It runs also much faster than the empirical tuning described by Devaine et al. [13] and used for the other rules which needs to run as many times the aggregation rule as the size of the parameter grid.

The ridge regression forecaster (Ridge) is presented in Algorithm 2. It was introduced in a stochastic setting by Hoerl and Kennard [19]. It forms at each instance the linear combination of experts minimizing a L_2-regularized least-square criterion on past data. It was first studied in the context of prediction with expert advice by Azoury and Warmuth [5] and Vovk [26] and was proved to enjoy nice theoretical properties, namely a regret bound $R_T = o(1)$ as $T \to \infty$ with respect to the best linear oracle. Once again, the learning parameter λ of the ridge regression aggregation rule has to be calibrated online. This tuning can be done using the methodology detailed in Devaine et al. [13, Section 2.4].

Ridge forms linear mixtures. The weights may be negative and not sum to one, while the other three aggregation rules restrict themselves to convex combination

Algorithm 2: The ridge regression forecaster (Ridge)

Parameter: $\lambda > 0$
Initialization: $\hat{\boldsymbol{p}}_0 = (1/K, \ldots, 1/K)$
For each instance $t = 1, 2, \ldots, T$

1. form the mixture $\hat{\boldsymbol{p}}_t$ defined by

$$\hat{\boldsymbol{p}}_t = \underset{\boldsymbol{u} \in \mathbb{R}^K}{\operatorname{argmin}} \left\{ \sum_{s=1}^{t-1} (y_s - \boldsymbol{u} \cdot \boldsymbol{x}_s)^2 + \lambda \left\| \boldsymbol{u} - \boldsymbol{p}_0 \right\|_2^2 \right\}$$

2. output prediction $\hat{y}_t = \hat{\boldsymbol{p}}_t \cdot \boldsymbol{x}_t$

of experts. In other words they only propose weight vectors $\hat{\boldsymbol{p}}_t \in \Delta_K$ where $\Delta_K = \{\boldsymbol{x} \in \mathbb{R}_+^K \ : \ \sum_i x_i = 1\}$. While linear aggregation rules might have more flexibility to detect correlation between experts and therefore often reach better performance, convex aggregation offers easy interpretation and safe predictions. Indeed convex weight vectors only assign non-negative weights to experts and their predictions always lie in the convex hull of experts predictions. Thus if all the experts are known to perform well, the aggregation rule will do so as well.

The gradient trick In the versions described above, EWA, FS, and ML-Poly compete only with the best fixed expert oracle. In Eq. (1) they cannot per se ensure vanishing average regret R_T with respect to the best fixed convex combination (i.e., $S = \Delta_K$). But it exists a standard reduction from the problem of competing with the best convex combination oracle to the goal of competing with the best fixed expert. This reduction is a well-known trick in the literature of individual sequences and is known as the *gradient trick*. The theoretical proof of this reduction is beyond the scope of this empirical research and is detailed in Cesa-bianchi and Lugosi [9, Section 2.5].

We only provide a brief description of the gradient trick. For each time instance t, we denote by $f_t : \boldsymbol{p} \in \Delta_K \mapsto \ell_t(\boldsymbol{p} \cdot \boldsymbol{x}_t) \in \mathbb{R}_+$ the function which evaluates the losses incurred by the weight vectors at time instance t. When the loss functions ℓ_t are convex and (sub)differentiable, the functions f_t are convex and (sub)differentiable over Δ_K. That is the case for instance for the square loss. We denote by ∇f_t the (sub)gradient function of f_t. The gradient trick relies then in not directly running the aggregation rule with the loss functions ℓ_t but with modified gradient loss functions $\tilde{f}_t : \boldsymbol{p} \in \Delta_K \mapsto \nabla f_t(\hat{\boldsymbol{p}}_t) \cdot \boldsymbol{p}$. In other words, the aggregation rules are run the same way by replacing the loss $\ell_t(\hat{y}_t)$ incurred by the algorithm by $\tilde{f}_t(\hat{\boldsymbol{p}}_t)$ and the loss $\ell_t(x_{k,t})$ suffered by expert k by $\tilde{f}_t(\delta_{k,t})$, where $\delta_k \in \Delta_K$ is the Dirac mass on k. Experiments of the next section are run using the gradient trick.

3 Experiments

We now describe the data we are dealing with and how we intend to build new experts from the three forecasting models described in the introduction. We then report the results obtained by mixing the different sets of experts as well as the performance of three reference oracles (best experts, best convex combination, best linear combination). As explained in Sect. 2 the performance of these oracles corresponds to the one aggregation rules hope to reach. Remember that the fixed share aggregation rule does not only compete with the best fixed convex combination but has a more ambitious goal. It aims at coming close to the performance of the best sequence of convex combinations that vary slowly enough. The results obtained by this more complex oracle will however not be reported in this research and we will only compare the performance of the fixed share aggregation rule to the best fixed convex combination of experts.

3.1 Presentation of the Data Set

We consider an electricity forecasting data set which corresponds to an updated version of the one analyzed by Devaine et al. [13]. It contains half-hourly measurements of the total electricity consumption of the EDF market in France from January 1, 2008 to June 15, 2012, together with several covariates, including temperature, cloud cover, wind, etc. Our goal is to forecast the consumption every day at 12:00 for the next 24 h; that is, for the next 48 time instances.

Atypical days are excluded from the data set. They correspond to public holidays as well as the days before and after them. Besides, the data set is cut into two subsets. A training set of 1,452 days from January 1, 2008 to August 31, 2011 is used to build the forecasting methods. The performance of the methods is then measured using the testing set of 244 days between September 1, 2011 to June 15, 2012. Prediction accuracy is measured in megawatts (MW) by the root mean squared error (RMSE)

$$\sqrt{\frac{1}{T}\sum_{t=1}^{T}(y_t-\hat{y}_t)^2}$$

and by the absolute percentage of error (MAPE)

$$\frac{1}{T}\sum_{t=1}^{T}\frac{|y_t-\hat{y}_t|}{y_t}. \tag{2}$$

Operational forecasting purposes require the predictions to be made simultaneously at 12:00 for the next 24 h (or equivalently for the next 48 half-hourly time

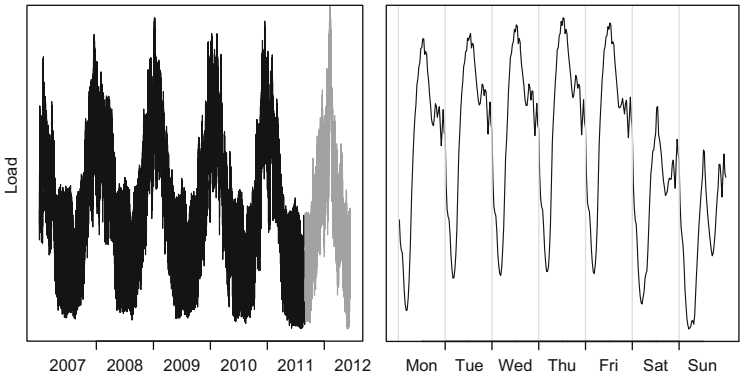

Fig. 1 (*left*) The observed half-hourly electricity consumptions between January 1, 2008 to June 15, 2012. An overall trend as well as a yearly seasonality can be pointed out in the data. The electrical heating in winter has a major impact in France on the electricity consumption. Approximately the last year is used to test the methods. (*right*) The observed half-hourly electricity consumptions during a typical week. A weekly pattern can be observed with a reduction of consumption during the week-end

instances) (Fig. 1). Aggregation rules can be adapted to this constraint via a generic extension detailed in Devaine et al. [13, Section 5.3].

3.2 Combining the Three Initial Models

From each of the three forecasting models described in the introduction, one expert is obtained: one from the generalized additive model (GAM), one from the curve linear regression (CLR) and a last one from the kernel approach based on wavelets (KWF). The experts are trained using the total training set from January 1, 2008 to August 31, 2011 described in the previous section. We calibrate the methods as presented in [4, 11, 25]. This starting set of three experts is denoted in the rest of the paper by E_0.

Table 1 reports the performance obtained by mixing the three experts in E_0. It describes also the reference results of the corresponding benchmark oracles: the best expert in E_0, the best convex combination and the best linear combination. The best convex combination and the best linear combination obtain similar results with RMSEs of 629 MW. Due to confidentiality constraints, we cannot provide detailed characteristics of the observed electricity consumptions. The relative performance of the methods can be enjoyed by noting that MAPEs are around 1 %. A significant improvement in performance can be noted in comparison to the best expert which obtains 744 MW. This motivates the necessity of mixing these models whose forecasts bring different information.

Table 1 Performance of oracles and aggregation rules using the set of experts E_0: GAM, CLR, and KWF

Oracles and aggregation rules	RMSE (MW)	MAPE (%)
Best expert	744	1.29
Best convex combination	629	1.06
Best linear combination	629	1.06
EWA	624	1.07
FS	625	1.05
ML-Poly	626	1.05
Ridge	638	1.06

EWA, FS, and ML-Poly are designed to compete with the best convex combination of experts while Ridge aims at approaching the performance of the best linear combination. The latter suffer RMSEs between 624 and 638 MW, which corresponds to reductions of the RMSE of approximatively 15 % compared to the best expert RMSE.

To quantify if our improvements are significant, we computed the dispersion of the errors among time instances of the aggregation rules and of the oracles—see technical report from Gaillard et al. [16, Section 1.2] for details. The dispersion is measured by the 95 % standard error

$$\hat{s}_t = \sqrt{\frac{\frac{1}{T}\sum_{t=1}^{T}\left((y_t-\hat{y}_t)^2 - \frac{1}{T}\sum_{t=1}^{T}(y_t-\hat{y}_t)^2\right)^2}{\frac{4}{T}\sum_{t=1}^{T}(y_t-\hat{y}_t)^2}},$$

that is, the half-width of the 95 % symmetric confidence interval of the error around the RMSEs reported in Tables 1–6. The 95 % standard error of the RMSEs are around 10 MW while the 95 % standard error of the MAPE are approximatively 0.02 %. Hence any reduction of the RMSE of more than 10 MW can be considered significant in the following.

Figure 2 reports the time evolution of the weights formed by ML-Poly and Ridge. The weight vectors created by Ridge converge but that is not obvious with ML-Poly. Stability is beneficial in an industrial context where weights have to be interpreted and understood by human beings. The weights formed by EWA and FS behave similarly to the ones of ML-Poly and are thus not reported here.

In the next section we will investigate how more experts can be designed based on these three models in order to improve further the predictions (Figs. 3 and 4).

3.3 Creating New Experts

We aim now at reducing the approximation error in Eq. (1), i.e., at improving the performance of the oracles, by adding new experts to our initial set E_0. If the new experts are not different enough from the base ones, the approximation term will

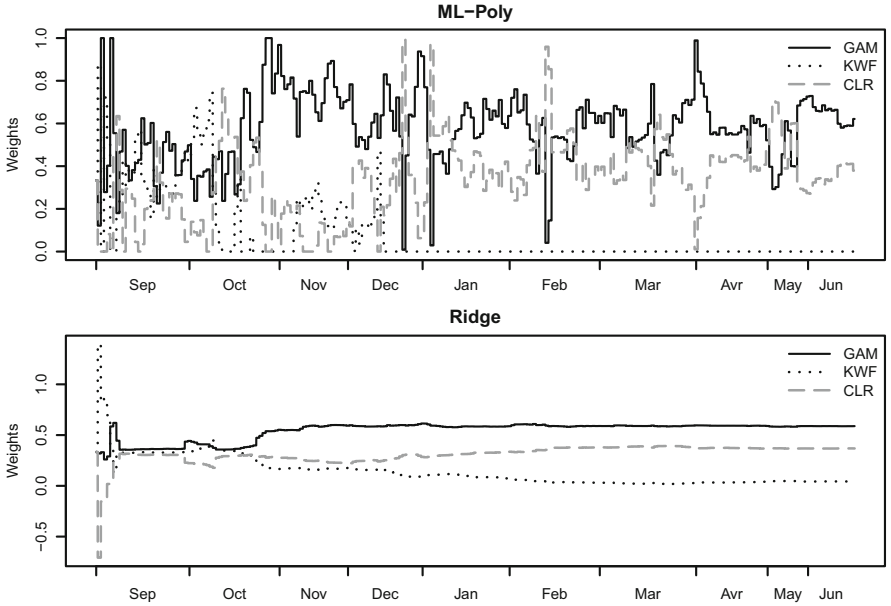

Fig. 2 Time evolution of the weight vectors formed by ML-Poly (*top*) and Ridge (*bottom*). We remark that the weights assigned by ML-Poly are always non-negative and sum to 1. Ridge can form negative weights

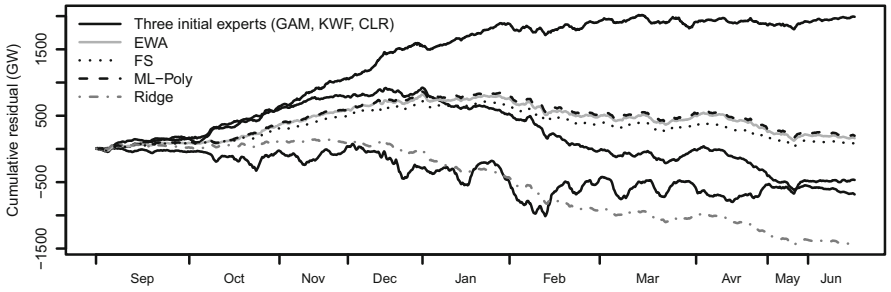

Fig. 3 Time evolution of cumulative residual of the three experts in E_0 and of the considered aggregation rules. The aggregation rules have smaller gradient in comparison to the experts. Besides it can be noticed that Ridge behaves very differently when compared to the other aggregation rules

not decrease; and worse, the right-most term in (1), the sequential estimation error, may increase, as the aggregation rule will have to face more experts. Note that none of the newly constructed experts will significantly outperform the performance of the best expert in E_0, which achieves a RMSE of 744 MW and a MAPE of 1.29 %. The benchmark performance of the best expert oracle thus remains the same for all considered extended sets of experts in this study.

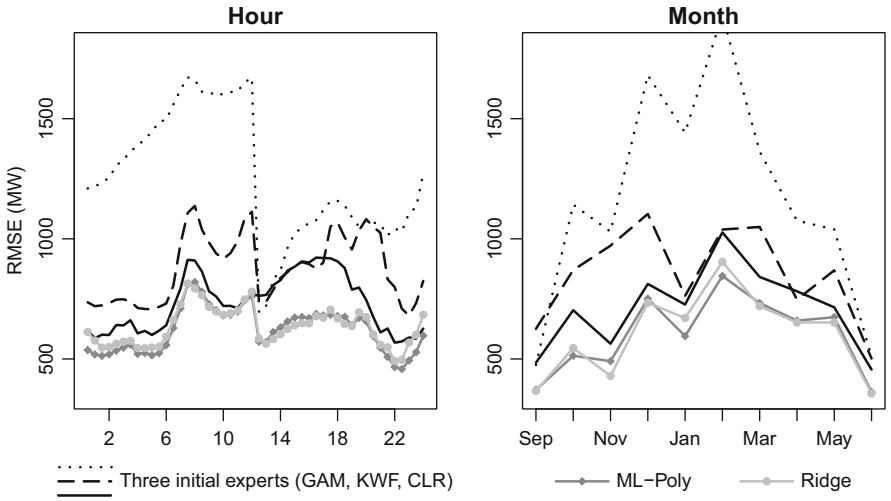

Fig. 4 Hourly and monthly RMSE of the first three experts and two aggregation rules described in Table 6. Because they obtain similar results to the ML-Poly aggregation rule, the EWA and the fixed share aggregation rules are not reported here

3.3.1 Bagging

The first method that we investigate is inspired from bagging, a machine learning method that combines bootstrapping with aggregating. It was introduced by Breiman [7] in order to improve the stability and the accuracy of a forecasting model. As most averaging methods it is known to reduce the variance and to avoid over-fitting. We aim at creating new experts by bootstrapping and at averaging online the newly constructed set of experts by running the aggregation rules.

Given a forecasting model, a bootstrapped expert is obtained by estimating the model on a random training strict subset S_0' (that does not include the whole training set S_0 of $n = 1{,}452$ days). The training set S_0' is generated by sampling n days from S_0 uniformly and with replacement. As the sampling is performed with replacement, some days may be present multiple times in S_0'. Breiman [7] pointed out that it leaves out $e^{-1} \approx 37\,\%$ of the days.

The bootstrap procedure is repeated 20 times using each of the three models at hand: GAM, CLR, and KWF. We name E_1 the set of 60 new experts, thus created. In our experiments we used 20 bootstrapped replicates of each model. This does not mean that more or fewer replicates would have led to worse performance. We wanted to add enough replicates to get sufficient variety but in the other hand we did not want to have too many bootstrapped experts in comparison to the experts we will build in the following sections. We tested several values and 20 expert replicates for each model seemed to be a reasonable trade off.

The performance of aggregation rules and oracles on $E_0 \cup E_1$ is reported in Table 2. The best linear combination oracle achieves a RMSE of 571 MW, which

Table 2 Performance of oracles and aggregation rules using the set of experts $E_0 \cup E_1$: GAM, CLR, KWF as well as the 60 bootstrapped experts

Oracles and aggregation rules	RMSE (MW)	MAPE (%)
Best convex combination	601	1.01
Best linear combination	571	0.99
EWA	614	1.01
FS	619	1.03
ML-Poly	612	1.02
Ridge	629	1.05

is a slightly better performance than the one of the best convex combination oracle, that equals 601 MW. This can be explained by two facts. First, the new experts might be biased. As their weights do not need to sum to one, linear mixtures correct better such bias. Second, as many experts are built using the same method, there are important correlations between them that can be better modeled using negative weights. However Ridge seems to have a hard time estimating the linear oracle and the performance is not much improved compared to Table 1. The empirical gain is about 10 MW for all aggregation rules. The improvement is thus not really significant.

3.3.2 Specialization

We start this section with the intuition that we need variety in our set of experts. We try to reuse the idea of bootstrapping to create new experts by modifying the training set. However, instead of sampling days uniformly in the training set E_0, we aim at assigning weights to training days with the goal to maximize the variety among themselves. To do so, we choose weights according to the values of the corresponding covariates (temperature, nebulosity, wind, type of day, ...). *Specialized experts* are created this way to some specific scenarios like heatwave, cold spell, sunny days or cloudy days. Hopefully if we choose different enough scenarios, these experts may catch different effects in the consumption that we might combine by aggregating them.

We now describe how to design such new experts. We suppose that we have at our disposal a forecasting model such that, during the training of the model, we can assign different weights to the elements of the training data. This is the case for GAM, CLR, and KWF for example. We assume that we also have access to an exogenous variable $Z \in [0, 1]$ like the temperature or the nebulosity which was normalized in $[0, 1]$. Given this model and this exogenous variable Z, we build two specialized experts: the first one by assigning to the day d the weight $(1 - Z_d)^2$, the second one with the choice Z_d^2. We thus get one expert focusing on high values of Z, and another one focusing on low values. The form of these weights was set empirically but we might want to replace it by many other forms. For instance, we had first looked at weights in $\{0, 1\}$ so as to select days according to a threshold on Z. However this led to unstable experts and poor performance. We

Table 3 Performance of oracles and aggregation rules using the set of experts $E_0 \cup E_2$: GAM, CLR, KWF as well as the 24 specialized experts

Oracles and aggregation rules	RMSE (MW)	MAPE (%)
Best convex combination	604	1.02
Best linear combination	582	0.99
EWA	609	1.01
FS	610	1.02
ML-Poly	602	1.00
Ridge	613	1.01

chose four covariables all based on temperature scenarios: the average, maximum, and minimum temperature of the day, and the variation of temperature with the previous day. We thus got 8 ($=$ 4 scenarios $\times$ 2 experts: hight and low) specialized experts by using each of the three models: GAM, CLR, and KWF. We call E_2 this set of 24 ($=$ 8 experts $\times$ 3 methods) experts. The performance obtained by mixing the experts in $E_0 \cup E_2$ is reported in Table 3. We observe a better performance for all aggregation rules with respect to bagging although we consider fewer additional experts.

Note that we showed the interest of specialized experts when they are combined with initial experts. The individual performance of specialized experts is often poor. They do not necessarily perform better than initial experts even when they are evaluated only on the data they should be specialized to.

3.3.3 Temporal Double-Scale Model

Now we study another way of constructing new experts by considering a temporal two-scale model. We follow the methodology detailed in Nedellec et al. [24] of the team TOLOLO for the "Kagle Global Energy Forecasting Competition 2012: Load Forecasting".

To forecast the short-term load with the canonical generalized additive model (GAM), the electricity consumption is usually explained by a single model including all the covariates (meteorological, and calendar ones) together with the recent consumption. The consumption Y_t is here decomposed into two parts: a medium-term part Y_t^{mt} including meteorological and calendar effects and a short-term part Y_t^{st} containing what could not be captured in large temporal scales, $Y_t = Y_t^{mt} + Y_t^{st}$. The short-term part Y_t^{st} basically consists of capturing local effects like extreme weather, network reconfigurations and so on. The modeling approach is thus divided into two estimation steps. First, we fit a mid-term generalized additive model including the meteorological and calendar covariates only. Second, we perform a residual analysis and we correct online the forecasts by using the observed consumptions of the last 30 days. This short-term readjusting is done by fitting another generalized additive model on the residuals.

The set containing this new expert is called E_3 and the performance obtained by combining this new expert with the three experts in E_0 is reported in Table 4. We

Table 4 Performance of oracles and aggregation rules using the set of experts $E_0 \cup E_3$: only four experts

Oracles and aggregation rules	RMSE (MW)	MAPE (%)
Best convex combination	596	1.00
Best linear combination	595	1.00
EWA	601	1.01
FS	599	1.00
ML-Poly	605	1.01
Ridge	605	1.00

observe RMSEs around 600 MW for all aggregation rules, which is a significant improvement considering that we add only one expert. The extension to other methods, like CLR and KWF, of this new way of creating experts is left for future work.

3.3.4 Boosting

In this section we investigate a final method to create new experts. We take now inspiration from boosting methods, like the AdaBoost algorithm of Freund and Schapire [15], that aims at correcting the mistakes of weak learners (or experts). The experts constructed in this section will be referred to as *boosted experts*.

Suppose that we have an expert that at an instance t of the training data estimates the consumption y_t by x_t. We want to build another expert predicting x'_t. Then reminding that our final aim is to aggregate well these predictions, it is irrelevant wether the second expert does not predict well y_t as soon as it counterbalances the error made by the original expert x_t. Improving the performance of the best convex combination should indeed only improve the prediction of the mixture. We can thus try to build the second expert so that the constant mixture $\gamma x_t + (1-\gamma)x'_t$ performs well for some $\gamma \in [0,1]$. This can be done by training the second experts not directly on the observed consumption y_t but on the modified one $y'_t = (y_t - \gamma x_t)/(1-\gamma)$. We can create several new experts by considering different values for $\gamma \in [0,1]$. Small values might lead to experts too similar from the original one, while larger values may create unstable experts.

We create 45 ($= 5 \times 3 \times 3$) new experts by using $\gamma \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$, each of the three initial experts in E_0 are used as the original expert x_t and each of the three models (GAM, CLR, and KWF) are used to create the modified experts x'_t. We denote by E_4 the set of 45 experts thus constructed.

We report in Table 5 the performance obtained by mixing experts in $E_0 \cup E_4$. We did not consider $\gamma < 0.5$ because the created experts were too similar to the original ones. Considering all $\gamma \in \{0.1, \ldots, 0.9\}$ does not affect the results (neither improve nor worsen them). The step size 0.1 of the grid was chosen arbitrarily and the investigation of different values is left for future research. The best convex combination oracle achieves a RMSE of 528 MW and the best linear combination oracle suffers a RMSE of 543 MW. The performance of *EWA* and *FS* is

Table 5 Performance of oracles and aggregation rules using the set of experts $E_0 \cup E_4$: GAM, CLR, KWF as well as the 45 boosted experts

Oracles and aggregation rules	RMSE (MW)	MAPE (%)
Best convex combination	543	0.93
Best linear combination	528	0.92
EWA	609	0.99
FS	609	0.99
ML-Poly	588	1.00
Ridge	578	0.98

Table 6 Performance of oracles and aggregation rules using the full set of experts $E_0 \cup E_1 \cup E_2 \cup E_4 \cup E_3$: all the 133 constructed experts

Oracles and aggregation rules	RMSE (MW)	MAPE (%)
Best convex combination	521	0.95
Best linear combination	479	0.84
EWA	578	0.95
FS	581	0.95
ML-Poly	565	0.95
Ridge	557	0.95

not much improved compared to previous experiments. They both incur RMSEs of 609 MW. But ML-Poly and Ridge suffer *rmse*s under 580 MW, which is a significant improvement.

3.3.5 Combining the Full Set of Experts

Table 6 reports the performance obtained by mixing all the experts created in the previous sections. We have now 133 experts at our disposal: 3 experts from in the starting set E_0, 60 bootstrapped experts in E_1, 24 specialized experts in E_2, 45 boosted experts in E_4 and 1 temporal two-scale model in E_3. The best linear combination and the best convex combination perform better. But at the same time it is harder to compete with them. Thus while the performance of aggregation rules is improved, the gap between oracles and aggregation rules is increased as well.

Ridge suffers in Table 6 a RMSE of 557 MW while it got 638 MW when mixing only the three experts in E_0 (see Table 1). The several refinement of the set of experts thus reduced its RMSE by approximatively 13 %. Similarly, the errors of EWA and FS were improved by about 7 % while ML-Poly got a 10 % reduction.

Figure 5 provides the RMSEs according to the number of experts aggregated with ML-Poly and Ridge. The experts included in the mixture were chosen by induction on the number of experts by following a forward approach. The induction was initialized with the expert which performed the best (744 MW). Suppose we had a set of K experts, the expert $K + 1$ was the one among the remaining experts that got the best results when it was mixed with the K experts using the considered rule. The procedure was stopped when all the 133 experts were used in the aggregation. The symbols in the figures represent the category (bootstrapped, specialized, boosting, etc.) of the last added expert.

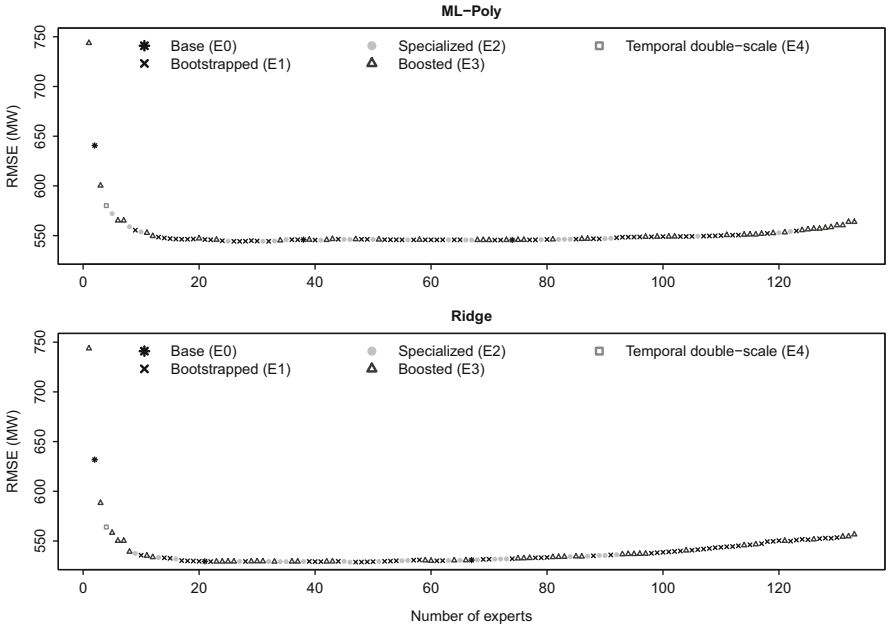

Fig. 5 Evolution of the performance according to the number of aggregated experts with ML-Poly (*top*) and Ridge (*bottom*)

Figure 5 shows the usual trade-off between having enough experts and overfitting. If we could select a good subset of experts to include in the mixture we could reduce the RMSE under the 530 MW bar by using Ridge (and approximatively under 545 MW by using ML-Poly). A suitable number of experts seems to lie between 15 and 90 experts. In future work, a pruning step, that would remove the less important experts before combining the forecasts of the remaining ones, might thus be a good option. Eban et al. [14] investigated in the framework of prediction of individual sequences a setting with many experts and few prediction instances. They remarked that trimming the worst experts often improves performance and suggested a procedure to do so online.

Note that the weights formed by ML-Poly and Ridge were different enough in Fig. 2. The aggregation rules might thus capture different information and we may thus try to combine them in a second layer. The simple uniform average of the forecasts of these two rules incurs a RMSE of 541 MW, while using one of the fancier sequential aggregation rules for the second layer gets us around 548 MW.

Figure 6 plots the hourly and monthly RMSEs of the two best aggregation rules and the RMSEs of the benchmark oracles described in Table 6. It shows that the aggregation rules always outperform in average the best single expert at all 48 half-hours of the day and at all 10 months of the testing set. In addition, we note a significant improvement of the performance at 12:30. This can be explained by the update of the weights, which occurs at noon. The best expert oracle, which is built

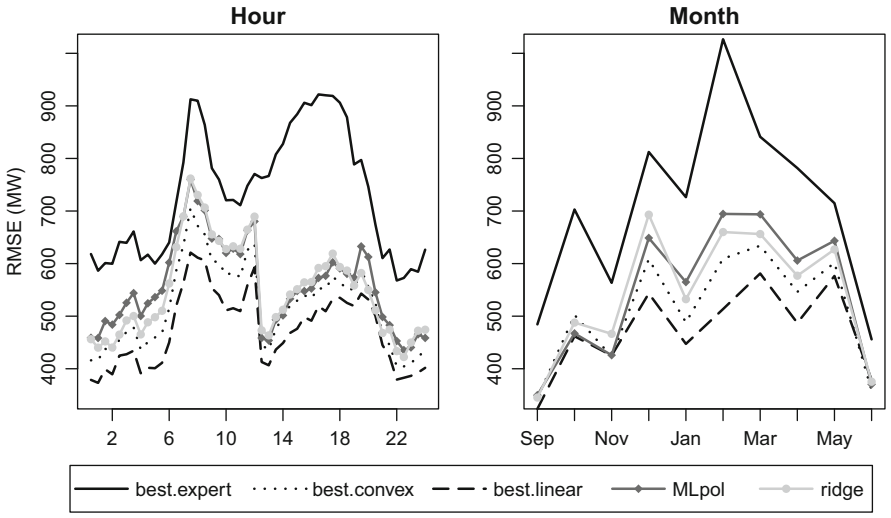

Fig. 6 Hourly and monthly RMSEs of the three benchmark oracles and of ML-Poly and Ridge described in Table 6

with a version of GAM, does not favor any hour of the day. The figure with monthly averaged RMSEs shows that aggregation rules do not only focus in improving forecasts when the task is easy. The best expert oracle is indeed outperformed every month, including November or February, which are month that are notoriously difficult to predict. Second, it illustrates that aggregation rules have a short learning period. They indeed encounter almost no regret during September and October with respect to all oracles although they just started to learn on September 1.

4 Conclusion

We presented in this paper an extensive application of aggregation rules from the literature of individual sequences to short-term electrical consumption forecasting. We focused on building an efficient set of experts from three initial ones, where the efficiency is viewed in terms of what these new experts bring to the combined forecasts. In other terms, we assumed that we had an efficient aggregation rule and focused more on reducing the approximation error, that is, the first term in (1). We noticed that despite the vast literature on combining forecasts (including empirical studies) rare papers dealt with this important topic. We proposed different strategies to generate experts from the three initial approaches: KWF, GAM, and CLR. We then quantified the gains in terms of forecast accuracy of the combined forecasts on the test set (about 10 month of half-hourly data). A summary of our results is presented in Fig. 7 for the two best aggregation rules: ML-Poly and Ridge.

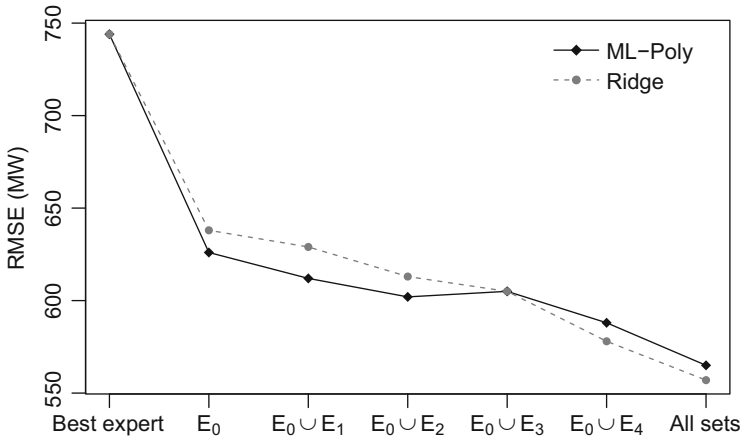

Fig. 7 RMSEs suffered by combining experts in E_0, $E_0 \cup E_1, \ldots, E_0 \cup E_4$ by using ML-Poly and Ridge. The performance of the best expert in E_0, and the final performance obtained by mixing all the experts in $E_0 \cup \cdots \cup E_4$ (referred to as 'All sets') are also reported

Combining all the experts that we generated with four different strategies, we achieved a 25 % gain over the best expert (around 200 MW in RMSE), which is a significant gain considering that the three original experts had already been refined and worked extremely well (they are not week learners as in classical boosting). This gain can be decomposed into two parts: roughly half of it comes from combining three heterogeneous initial experts, the other half is due to the construction of new experts. Among the four proposed strategies, our boosting trick and what we call specialized experts bring the most important improvements. We believe that these strategies could be applied to other forecasting problems and there is still some work to derive theoretical guarantees for these tricks. We also observe that aggregating rules are quite robust to adding new experts, and it is clear in Fig. 5 that combining forecasts does not suffer much from over fitting. Nevertheless, these results suggest that there is a way for improving the aggregation rules accuracy by adding a pruning step that could select the best set of experts in some online fashion.

Acknowledgements We thank the anonymous reviewers, the editors, and Gilles Stoltz for their valuable comments and feedback.

References

1. Aiolfi, M., Capistrán, C., & Timmermann, A. (2010). *Forecast combinations* (Working Papers 2010-04). Banco de México. http://EconPapers.repec.org/RePEc:bdm:wpaper:2010-04.
2. Antoniadis, A., Brossat, X., Cugliari, J., & Poggi, J. (2012). Prévision d'un processus à valeurs fonctionnelles en présence de non stationnarités. Application à la consommation d'électricité. *Journal de la Société FranÃČÂğaise de Statistique, 153*(2), 52–78.
3. Antoniadis, A., Brossat, X., Cugliari, J., & Poggi, J. (2013). Clustering functional data using wavelets. *International Journal of Wavelets, Multiresolution and Information Processing, 11*(1), 1–30.
4. Antoniadis, A., Paparoditis, E., & Sapatinas, T. (2006). A functional wavelet–kernel approach for time series prediction. *Journal of the Royal Statistical Society: Series B, 68*(5), 837–857.
5. Azoury, K. S., & Warmuth, M. K. (2001). Relative loss bounds for on-line density estimation with the exponential family of distributions. *Machine Learning, 43*(3), 211–246.
6. Biau, G., & Patra, B. (2011). Sequential quantile prediction of time series. *IEEE Transactions on Information Theory, 57*(3), 1664–1674.
7. Breiman, L. (1996). Bagging predictor. *Machine Learning, 24*(2), 123–140.
8. Cesa-Bianchi, N., & Lugosi, G. (2003). Potential-based algorithms in on-line prediction and game theory. *Machine Learning, 51*(3), 239–261.
9. Cesa-Bianchi, N., & Lugosi, G. (2006). *Prediction, learning, and games*. Cambridge/ New York: Cambridge University Press.
10. Cho, H., Goude, Y., Brossat, X., & Yao, Q. (2013). Modeling and forecasting daily electricity load curves: A hybrid approach. *Journal of the American Statistical Association, 108*, 7–21.
11. Cho, H., Goude, Y., Brossat, X., & Yao, Q. (2014, to appear). Modeling and forecasting daily electricity load using curve linear regression. In *Lecture notes in statistics 217: Modeling and stochastic learning for forecasting in high dimension*, 35–52.
12. Clemen, R. T. (1989). Combining forecasts: A review and annotated bibliography. *International Journal of Forecasting, 5*(4), 559–583.
13. Devaine, M., Gaillard, P., Goude, Y., & Stoltz, G. (2013). Forecasting electricity consumption by aggregating specialized experts. *Machine Learning, 90*(2), 231–260.
14. Eban, E., Birnbaum, A., Shalev-Shwartz, S., & Globerson, A. (2012). Learning the experts for online sequence prediction. In *Proceedings of ICML*, Edinburgh.
15. Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an applicationto boosting. *Journal of Computer and System Sciences, 55*, 119–139.
16. Gaillard, P., Goude, Y., & Stoltz, G. (2011). *A further look at the forecasting of the electricity consumption by aggregation of specialized experts* (Technical report). pierre.gaillard.me/doc/GaGoSt-report.pdf.
17. Gaillard, P., Stoltz, G., & van Erven, T. (2014). A second-order bound with excess losses. ArXiv:1402.2044.
18. Herbster, M., & Warmuth, M. K. (1998). Tracking the best expert. *Machine Learning, 32*(2), 151–178.
19. Hoerl, A., & Kennard, R. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics, 12*, 55–67.
20. Littlestone, N., & Warmuth, M. K. (1994). The weighted majority algorithm. *Information and Computation, 108*(2), 212–261.
21. Mallet, V. (2010). Ensemble forecast of analyses: Coupling data assimilation and sequential aggregation. *Journal of Geophysical Research, 115*(D24303), 1–10.
22. Mallet, V., Stoltz, G., & Mauricette, B. (2009). Ozone ensemble forecast with machine learning algorithms. *Journal of Geophysical Research, 114*(D05307), 1–13.
23. Monteleoni, C., Schmidt, G. A., Saroha, S., & Asplund, E. (2011). Tracking climate models. *Statistical Analysis and Data Mining, 4*(4), 372–392.

24. Nedellec, R., Cugliari, J., & Goude, Y. (2014). Gefcom2012: Electric load forecasting and backcasting with semi-parametric models. *International Journal of Forecasting, 30*(2), 375–381.
25. Pierrot, A., & Goude, Y. (2011). Short-term electricity load forecasting with generalized additive models. In: *Proceedings of ISAP power*, Hersonisos, Greece (pp. 593–600).
26. Vovk, V. (2001). Competitive on-line statistics. *International Statistical Review, 69*(2), 213–248.
27. Vovk, V. G. (1990). Aggregating strategies. In *Proceedings of the Third Workshop on Computational Learning Theory*, Rochester (pp. 371–386).
28. Wood, S. (2006). *Generalized additive models: An introduction with R*. Boca Raton: Chapman and Hall/CRC.
29. Wood, S., Goude, Y., & Shaw, S. (2015). Generalized additive models for large datasets. *Journal of Royal Statistical Society, Series C, 64*(1), 139–155.