A further look at the forecasting of the electricity consumption by aggregation of specialized experts

Pierre Gaillard Yannig Goude Gilles Stoltz

February 3, 2012

1. Overview of the setting

1.1. Notations

We consider the sequential prediction of arbitrary individual sequences based on expert advice. This paper is based on methods originally presented in [**?**] for the setting of electricity forcasting.

We have a set $E = \{1, \dots, N\}$ of experts at our disposal. At each instance $t = 1, \dots, T$, some experts are active and output predictions in a convex outcome space $\mathcal{Y}$, typically $\mathbb{R}_+$. We denote by $E_t \subset E$ the set of active experts and by f_{it} the prediction of expert i if $i \in E_t$. An aggregation rule $\mathcal{A}$ then forms a mixture $\boldsymbol{p}_t = (p_{1t}, \dots, p_{Nt}) \in \mathbb{R}^N$. Its prediction is given by

$$\widehat{y}_t = \sum_{i \in E_t} p_{it} f_{it} \, .$$

The realized consumption y_t is then revealed and instance $t+1$ starts.

We often restrict the prediction to convex weight vectors. That is, $\boldsymbol{p}_t \in \mathcal{X}_{E_t}$ where $\mathcal{X}_{E_t}$ is the subset of $\mathbb{R}^N$ where for all $i \in E$, $p_{it} \geqslant 0$; $\sum_{j \in E_t} p_{jt} = 1$ and $\sum_{j \notin E_t} p_{jt} = 0$. $\mathcal{X}$ denotes $\mathcal{X}_E$.

1.2. Assessment of the quality of a sequence of predictions

To measure the accuracy of the prediction $\widehat{y}_t$ proposed at round t for the observation y_t we consider a loss function $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$. At each time instance t, the mixture $\boldsymbol{p}_t$ output by the rule is thus evaluated by the loss function $\ell_t : \mathcal{X} \to \mathbb{R}$ defined by

$$\ell_t(\boldsymbol{p}) = \ell\left(\sum_{j \in E_t} p_j f_{jt}, \, y_t \right)$$

for all $\boldsymbol{p} \in \mathcal{X}$. Our goal is to design sequential aggregation rules $\mathcal{A}$ with a small average error,

$$\overline{\mathrm{ERR}}_T(\mathcal{A}) = \frac{1}{T} \sum_{t=1}^{T} \ell_t(\boldsymbol{p}_t) \, .$$

Considered loss functions

In our experiments we used three different loss functions. Typical aggregation rules put more weight on more accurate experts, therefore their definition depends on the losses suffered by the experts in the past. We will thus have three versions of each algorithm depending on the loss function used to assess the quality of experts. The performance of each version will then be characterized by its average error and a corresponding measure of dispersion, which may depend on the specific loss function at hand.

- *The square loss* is defined for all $x, y \in \mathbb{R}_+$ by

$$\ell(x, y) = (x - y)^2 \, .$$

In this case, instead of $\overline{\mathrm{ERR}}_T(\mathcal{A})$ we will use the root mean square error

$$\mathrm{RMSE}_T(\mathcal{A}) = \sqrt{\overline{\mathrm{ERR}}_T(\mathcal{A})} = \sqrt{\frac{1}{T} \sum_{t=1}^{T} \left(\widehat{y}_t - y\right)^2}$$

to report the quality of aggregation rule $\mathcal{A}$ according to the square loss. We can get the corresponding dispersion with the delta method and Slutsky's lemma (cf. Appendix A),

$$\widehat{s}_T = \sqrt{\frac{\frac{1}{T}\sum_{t=1}^{T}\left(\left(\widehat{y}_t - y_t\right)^2 - \frac{1}{T}\sum_{t'=1}^{T}\left(\widehat{y}_{t'} - y_{t'}\right)^2\right)^2}{4\,\frac{1}{T}\sum_{t=1}^{T}\left(\widehat{y}_t - y_t\right)^2}}\,.$$

We report the 95% standard error $1.96\ \widehat{s}_T/\sqrt{T}$ in the tables.

- *The absolute error* is defined for all $x, y \in \mathbb{R}_+$ by

$$\ell(x,y) = |x-y|\,.$$

For this loss function, since we use

$$\textsc{mae}_T(\mathcal{A}) = \frac{1}{T}\sum_{t=1}^{T}\left|\widehat{y}_t - y_t\right|,$$

to quantify the quality of an algorithm $\mathcal{A}$, the corresponding measure of dispersion is defined as the standard deviation of the sample,

$$\widehat{\sigma}_T = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(\left|\widehat{y}_t - y_t\right| - \frac{1}{T}\sum_{t'=1}^{T}\left|\widehat{y}_{t'} - y_{t'}\right|\right)^2}\,.$$

We report the 95% standard error $1.96\ \widehat{\sigma}_T/\sqrt{T}$ in the tables.

- *The absolute percentage of error* is defined for all $x, y \in \mathbb{R}_+$ by

$$\ell(x,y) = \frac{|x-y|}{y}\,.$$

We use

$$\textsc{mape}_T(\mathcal{A}) = \frac{1}{T}\sum_{t=1}^{T}\frac{\left|\widehat{y}_t - y_t\right|}{y_t}$$

to measure the error and the standard deviation of the sample,

$$\widehat{\sigma}_T = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(\frac{\left|\widehat{y}_t - y_t\right|}{y_t} - \frac{1}{T}\sum_{t'=1}^{T}\frac{\left|\widehat{y}_{t'} - y_{t'}\right|}{y_{t'}}\right)^2},$$

to quantify the dispersion. We then report the 95% standard error $1.96\ \widehat{\sigma}_T/\sqrt{T}$ in the tables.

- *The correlation* is defined for an aggregation rule $\mathcal{A}$ outputting predictions $\widehat{y}_1, \ldots, \widehat{y}_T$ as

$$\textsc{corr}_T(\mathcal{A}) = \frac{\sum_{t=1}^{T}\left(\widehat{y}_t - \overline{\widehat{y}}\right)\left(y_t - \overline{y}\right)}{\sqrt{\sum_{t=1}^{T}\left(\widehat{y}_t - \overline{\widehat{y}}\right)^2\sum_{t=1}^{T}\left(y_t - \overline{y}\right)^2}},$$

where $\overline{y} = \frac{1}{T}\sum_{t=1}^{T} y_t$ and $\overline{\widehat{y}} = \frac{1}{T}\sum_{t=1}^{T}\widehat{y}_t$.

Reference oracles

Intuitively, if all experts are poor, there is no reason for our aggregation rule to get good results. Hence, to evaluate the performance of our algorithms, we will compare their errors with the errors of some so-called oracles. We present here the reference oracles that we considered and how they are defined in the framework of specialized experts.

In the following, we define average errors $\overline{\mathrm{ERR}}_T$ (as above, we will rather consider $\mathrm{RMSE}_T = \sqrt{\overline{\mathrm{ERR}}_T}$ for the square loss).

- *Fixed expert.* We denote by δ_i the rule that always follows the prediction of expert i. Since it is not well defined on all time instances, we only evaluate it on the instances when expert i is active,

$$\overline{\mathrm{ERR}}_T(\delta_i) = \frac{1}{\sum_{t=1}^T \mathbb{1}_{\{i \in E_t\}}} \sum_{t=1}^T \ell(f_{it}, y_t)\, \mathbb{1}_{\{i \in E_t\}} \,.$$

  The best fixed expert oracle $\mathcal{O}_\delta$ is then defined as

$$\mathcal{O}_\delta \in \arg\min_{\delta_i} \overline{\mathrm{ERR}}^1_T(\delta_i) \,.$$

- *Fixed linear combination (definition 1).* It corresponds to the use of a fixed linear weight vector $\boldsymbol{u} \in \mathbb{R}^N$, to be renormalized at each time instance so that it puts a probability mass of 1 on E_t. Formally, we generalize the definition of $\overline{\mathrm{ERR}}_T$ for all $\boldsymbol{u} \in \mathbb{R}^N$ as

$$\overline{\mathrm{ERR}}^1_T(\boldsymbol{u}) = \frac{1}{\sum_{t=1}^T \big|\boldsymbol{u}(E_t)\big|} \sum_{t=1}^T \ell_t\left(\frac{\boldsymbol{u}}{\boldsymbol{u}(E_t)}\right) \big|\boldsymbol{u}(E_t)\big| \,,$$

  where $\boldsymbol{u}(E_t) = \sum_{j \in E_t} u_j$.

  Note that we retrieve the previous definition for $\boldsymbol{u} = \delta_i$. The best fixed linear combination (version 1) is then defined as

$$\mathcal{O}^1_{\mathbb{R}^N} \in \arg\min_{\boldsymbol{u} \in \mathbb{R}^N} \overline{\mathrm{ERR}}^1_T(\boldsymbol{u}) \,.$$

- *Fixed linear combination (definition 2).* This definition is similar to the previous one, with the exception of a small twist in the normalization:

$$\overline{\mathrm{ERR}}^2_T(\boldsymbol{u}) = \frac{1}{\sum_{t=1}^T \big|\tau_t(\boldsymbol{u})\big|} \sum_{t=1}^T \ell_t\left(\frac{\boldsymbol{u}}{\tau_t(\boldsymbol{u})}\right) \big|\tau_t(\boldsymbol{u})\big| \,,$$

  where $\tau_t(\boldsymbol{u}) = \dfrac{\sum_{j \in E_t} u_j}{\sum_{k=1}^N u_k}$.

  The best fixed linear combination (version 2) is then defined as

$$\mathcal{O}^2_{\mathbb{R}^N} \in \arg\min_{\boldsymbol{u} \in \mathbb{R}^N} \overline{\mathrm{ERR}}^2_T(\boldsymbol{u}) \,.$$

  Since Definition 1 is stable by homothetical changes (i.e., $\overline{\mathrm{ERR}}^1(\boldsymbol{u}) = \overline{\mathrm{ERR}}^1(\lambda \boldsymbol{u})$), it is easy to see that $\min_{\boldsymbol{u}} \overline{\mathrm{ERR}}^2 \leqslant \min_{\boldsymbol{u}} \overline{\mathrm{ERR}}^1_{\boldsymbol{u}}$. However, in the simulations the same performance was obtained for both oracles (this might be caused by the optimization technique used to compute the oracles). This is why we report only a single such oracle value in the tables.

- *Fixed convex combination.* It is a special case of the previous oracle (version 2) with an additional convex constraint on the weight vectors:

$$\overline{\mathrm{ERR}}_T(\boldsymbol{q}) = \frac{1}{\sum_{t=1}^T \big|\tau_t(\boldsymbol{q})\big|} \sum_{t=1}^T \ell_t\left(\frac{\boldsymbol{q}}{\tau_t(\boldsymbol{q})}\right) \big|\tau_t(\boldsymbol{q})\big| \,,$$

  where $\tau_t(\boldsymbol{q}) = \sum_{j \in E_t} q_j$. The best fixed convex combination is given by

$$\mathcal{O}_{\mathcal{X}} \in \operatorname*{arg\,min}_{\boldsymbol{q} \in \mathcal{X}} \overline{\mathrm{ERR}}_T^2(\boldsymbol{q}) \,.$$

- *Sequences of experts with few shifts.* The activations and deactivations of experts prevent the aggregation rules from picking a constant expert over time. We therefore authorize a few shifts and consider the legal sequences with at most m shifts,

$$\mathcal{L}_\delta^m = \left\{ (\delta_{i_1}, \ldots, \delta_{i_T}) \,\middle|\, \forall t \,, \quad i_t \in E_t \ \text{ and } \ \#\{t,\ i_t \neq i_{t+1}\} \leqslant m \right\}.$$

  We may define the best sequence of experts with at most m shifts as

$$\mathcal{O}_\delta^m \in \operatorname*{arg\,min}_{\mathcal{A} \in \mathcal{L}_\delta^m} \overline{\mathrm{ERR}}_T(\mathcal{A}) \,.$$

- *Sequences of convex combinations of experts with few shifts.* It is a generalization of the previous oracle to convex combinations. We define the set of legal sequences as

$$\mathcal{L}_{\mathcal{X}}^m = \left\{ (\boldsymbol{q}_1, \ldots, \boldsymbol{q}_T) \,\middle|\, \forall t \,, \quad \boldsymbol{q}_t \in \mathcal{X}_{E_t} \ \text{ and } \ \#\{t,\ \boldsymbol{q}_t \neq \boldsymbol{q}_{t+1}\} \leqslant m \right\}.$$

  The oracle is then given by

$$\mathcal{O}_{\mathcal{X}}^m \in \operatorname*{arg\,min}_{\mathcal{A} \in \mathcal{L}_{\mathcal{X}}^m} \overline{\mathrm{ERR}}_T(\mathcal{A}) \,.$$

Reference aggregation rules

We also consider two simple reference aggregation rules which will be used as benchmarks.

- *Uniform mixture.* We denote by $\mathcal{U}_m$ this strategy. It forms a uniform average of the predictions of active experts,

$$\widehat{y}_t(\mathcal{U}_m) = \frac{1}{|E_t|} \sum_{i \in E_t} f_{it} \,.$$

- *Uniform convex weight vector.* We denote by $\mathcal{U}_c$ this strategy. It corresponds to the use of the uniform combination $\mathbf{1}_N = \big(1/N, \ldots, 1/N\big)$. Note that it outputs the same predictions as the uniform mixture, but its performance is assessed in a different manner; indeed,

$$\overline{\mathrm{ERR}}_T(\mathcal{U}_m) = \frac{1}{T} \sum_{t=1}^T \ell\left(\frac{1}{|E_t|} \sum_{i \in E_t} f_{it},\, y_t \right) ,$$

  while

$$\overline{\mathrm{ERR}}_T(\mathcal{U}_c) = \frac{1}{\sum_{t=1}^T \mathbf{1}_N(E_t)} \sum_{t=1}^T \ell\left(\frac{1}{|E_t|} \sum_{i \in E_t} f_{it},\, y_t \right) \mathbf{1}_N(E_T) \,.$$

2. Experiments

2.1. Description of the data set

It consists in half-hourly observations of the French electricity consumption from September 1, 2007 to August 31, 2008 (henceforth referred to as the prediction set). We have 24 experts at our disposal; they can be grouped into three families. The units are in Gigawatts (GW). Since the square loss and the absolute loss are not scale-independent, the parameters may depend on the unit chosen for these two loss functions (in contrast to what happens for the absolute percentage of error).

Number of days D	320
Time intervals	Every 30 minutes
Time instances T	15 360 ($= 320 \times 48$)
Number of experts N	24 ($= 15 + 8 + 1$)
Unit	GW
Median of the y_t	56.33
Bound B on the y_t	92.76

Table 1: Some characteristics of the observations y_t (half-hourly mean consumptions) of the considered data set.

Considered experts

The experts have been constructed thanks to a training set formed by pairs of realized energy consumptions and of some contextual variables observed during a given period of time before the one corresponding to the prediction set. The size and the nature of the training set may depend on the expert. The three families of considered experts are listed below.

- *Parametric model.* Implemented in EDF R&D prediction system as Eventail. By changing the parameters we got 15 experts in this family.
- *Semi-parametric model.* Generative additive model (GAM). We obtained 8 experts.
- *Functional model.* 1 expert.

Reference performance

In Table 2, we report the performance of the benchmark procedures. The performance of the best convex and linear weight vectors are obtained by repeatedly performing an algorithm of Byrd et al. [**?**], a local optimization method which allows box constraints; initial values are sampled uniformly at random in $\mathcal{X}$ at the beginning of each new attempt (wether convex or linear oracles are to be computed). It is implemented in `R` with the command `optim` in combination with the parameter "`L-BFGS-B`." The performance of the best compound expert is obtained by dynamic programming; see Appendix B for detailed explanations.

Benchmark procedure	Optimized in	RMSE $\times 10^3$	MAE $\times 10^3$	MAPE $\times 10^2$	CORR %
Uniform mixture $\mathcal{U}_m$		724 ± 11	545 ± 7	0.960 ± 0.012	99.8
Uniform convex weight vector $\mathcal{U}_c$		748 ± 11	564 ± 7	0.960 ± 0.012	99.8
Best single expert $\mathcal{O}_\delta$	ALL[a]	782 ± 10	602 ± 8	1.050 ± 0.010	99.2
Best convex weight vector $\mathcal{O}_\mathcal{X}$	RMSE	658 ± 9	501 ± 6	0.875 ± 0.010	99.8
	MAE	660 ± 14	500 ± 10	0.872 ± 0.017	99.8
	MAPE	662 ± 9	502 ± 7	0.868 ± 0.011	99.8
Best linear weight vector $\mathcal{O}_{\mathbb{R}^N}$ [b]	RMSE	625 ± 7	481 ± 5	0.857 ± 0.009	99.8
	MAE	627 ± 7	480 ± 5	0.856 ± 0.010	99.8
	MAPE	633 ± 8	484 ± 5	0.848 ± 0.009	99.8
Best compound expert $\mathcal{O}_\delta^m$	RMSE				
Size at most $m = 50$		$534 \pm .$	$416 \pm .$	$0.745 \pm .$	.
Size at most $m = 200$		$414 \pm .$	$319 \pm .$	$0.573 \pm .$	.
Size at most $m = T - 1 = 15\,359$		$223 \pm .$	$118 \pm .$	$0.211 \pm .$	.
Best compound expert $\mathcal{O}_\delta^m$	MAE				
Size at most $m = 50$		$541 \pm .$	$412 \pm .$	$0.735 \pm .$	.
Size at most $m = 200$		$418 \pm .$	$316 \pm .$	$0.566 \pm .$	.
Size at most $m = T - 1 = 15\,359$		$223 \pm .$	$118 \pm .$	$0.211 \pm .$	.
Best compound expert $\mathcal{O}_\delta^m$	MAPE				
Size at most $m = 50$		$545 \pm .$	$413 \pm .$	$0.734 \pm .$	.
Size at most $m = 200$		$421 \pm .$	$317 \pm .$	$0.563 \pm .$	.
Size at most $m = T - 1 = 15\,359$		$223 \pm .$	$118 \pm .$	$0.211 \pm .$	.

[a] The same expert achieves the best performance for all loss functions.
[b] We recall that $\mathcal{O}^1_{\mathbb{R}^N}$ and $\mathcal{O}^2_{\mathbb{R}^N}$ reach similar performance on this data set.

Table 2: Performance of reference oracles and strategies on the data set.

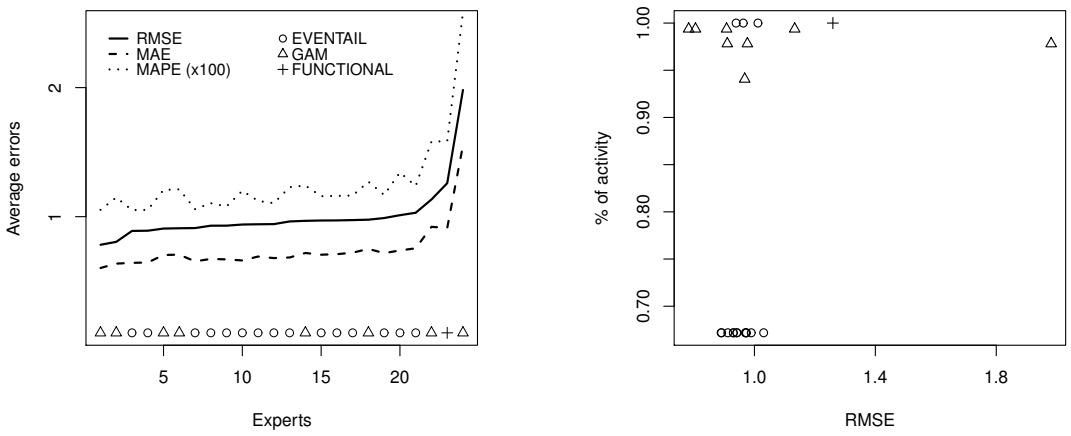


Figure 1: [Left figure] Average errors and RMSEs of the experts (y-axis) sorted according to their RMSEs (x-axis). [Right figure] Frequencies of activity of the experts (y-axis) according to their RMSEs (x-axis). The Eventail experts are indexed by ◯, the GAM experts by △, and the functional expert by +.

2.2. Performance of the considered aggregation rules

The considered aggregation rules can be clustered into three families: exponentially weighted average (EWA), specialist, and fixed-share aggregation rules. We respectively denote by $\mathcal{W}_\eta$, $\mathcal{S}_\eta$, and $\mathcal{F}_{\eta\alpha}$ their basic versions and by $\mathcal{W}_\eta^{\text{grad}}$, $\mathcal{S}_\eta^{\text{grad}}$, and $\mathcal{F}_{\eta\alpha}^{\text{grad}}$ their gradient versions. For more details on the aggregation rules, the reader is referred to [**?**].

Adaptation to an operational constraint

In [**?**], EWA and fixed-share aggregation rules are implemented with an operational constraint. It consists in forecasting simultaneously every day at 12:00 the next 48 time instances. We extended the specialist and EWA aggregation rules to this constraint in a generic manner (cf. Algorithm 1). For fixed-share aggregation rules, we kept the extension proposed in [**?**]. In the following, we always deal with the operational extensions of the aggregation rules and we keep the previous notations to denote them.

Algorithm 1 Extension of an aggregation rule $\mathcal{A}$ to operational forecasting.

`Input:` aggregation rule $\mathcal{A}$
`Initialization:` uniform initial weight vector $\boldsymbol{w}_1 \in \mathcal{X}$

`for` instance t `from` 1 `to` T `do`
  `predict` $\widehat{y}_t \leftarrow \dfrac{1}{\sum_{i\in E_t} w_{it}} \sum_{j\in E_t} w_{jt} f_{jt}$
  `if` $t = 48k$ `for some` k
    $\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{p}_{t+1}(\mathcal{A})$ // synchronize, see footnote [a]
  `else`
    $\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t$ // do not update
  `end if`
`end for`

[a] $\boldsymbol{p}_{t+1}(\mathcal{A})$ is the convex weight vector chosen by $\mathcal{A}$ after observing $y_1, \ldots, y_t$ and the corresponding experts predictions

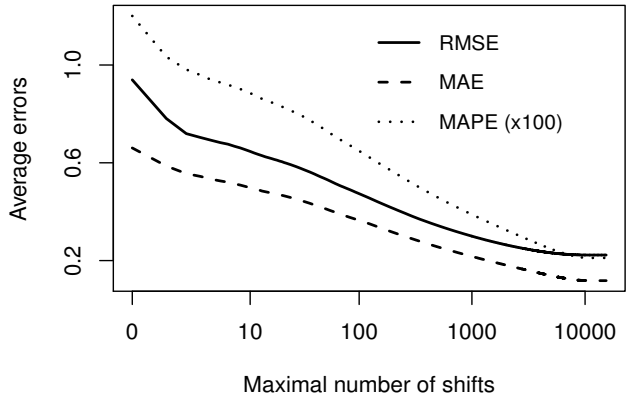


Figure 2: Evolution of the average errors of the best compound experts $\mathcal{O}_\delta^m$ (y-axis) according to the number of shifts m (x-axis).

Performance for constant parameters

In Tables 3, 5, and 4, we report the performance of the considered aggregation rules for constant parameters. We initialized the weights with a uniform distribution on the active experts. The parameters are optimized on a grid, according to the best performance obtained with the loss function used to build the rule.

Table 3 (resp., Table 4) describes the performance of $\mathcal{W}_\eta$ and $\mathcal{W}_\eta^{\text{grad}}$ (resp., of $\mathcal{S}_\eta$ and $\mathcal{S}_\eta^{\text{grad}}$) for the best constant parameter η on the grid

$$\Lambda = \left\{ m \cdot 10^k, \quad m \in \{1, \dots, 9\} \text{ and } k \in \{-7, \dots, 1\} \right\}.$$

The performance of $\mathcal{W}_\eta$ (resp., $\mathcal{W}_\eta^{\text{grad}}$) should be compared to the one of the best fixed expert $\mathcal{O}_\delta$ (resp., of the best convex weight vector $\mathcal{O}_\mathcal{X}$).

Table 5 reports the errors of fixed-share type algorithms for the best constant pairs of parameters (η, α) on the grid

$$\Lambda_\mathcal{F} = \left\{ \left(m \cdot 10^k, \alpha\right), \ m \in \{1, \dots, 9\}, \ k \in \{-5, \dots, 3\}, \text{ and } \alpha \in \{0, 0.005, 0.01, 0.05, 0.1, 0.2, 0.5, 1\} \right\}.$$

Aggregation rule	Version	Best η	RMSE $\times 10^3$	MAE $\times 10^3$	MAPE $\times 10^2$	CORR %
$\mathcal{W}_\eta$	RMSE	$1 \cdot 10^{-4}$	718 ± 12	539 ± 7	0.953 ± 0.012	99.8
	MAE	$2 \cdot 10^{-4}$	721 ± 11	541 ± 7	0.956 ± 0.012	99.8
	MAPE	$9 \cdot 10^{-3}$	720 ± 12	541 ± 7	0.955 ± 0.012	99.8
$\mathcal{W}_\eta^{\text{grad}}$	RMSE	2	629 ± 8	483 ± 6	0.859 ± 0.011	99.8
	MAE	$4 \cdot 10^{-2}$	629 ± 8	481 ± 6	0.857 ± 0.011	99.8
	MAPE	3	631 ± 9	481 ± 6	0.857 ± 0.011	99.8

Table 3: Performance of $\mathcal{W}$ and $\mathcal{W}_{\text{grad}}$ with the best constant parameters on the grid Λ.

Aggregation rule	Version	Best η	RMSE $\times 10^3$	MAE $\times 10^3$	MAPE $\times 10^2$	CORR %
$\mathcal{S}_\eta$	RMSE	$1 \cdot 10^{-4}$	718 ± 12	539 ± 7	0.953 ± 0.013	99.8
	MAE	$4 \cdot 10^{1}$	687 ± 10	512 ± 7	0.912 ± 0.012	99.8
	MAPE	$8 \cdot 10^{-3}$	720 ± 12	541 ± 7	0.955 ± 0.012	99.8
$\mathcal{S}_\eta^{\text{grad}}$	RMSE	$3 \cdot 10^{-2}$	631 ± 9	482 ± 8	0.861 ± 0.011	99.8
	MAE	$4 \cdot 10^{-2}$	630 ± 8	481 ± 6	0.858 ± 0.011	99.8
	MAPE	2	630 ± 8	481 ± 6	0.856 ± 0.011	99.8

Table 4: Performance of $\mathcal{S}_\eta$ and $\mathcal{S}_\eta^{\text{grad}}$ with the best constant parameters on the grid Λ.

Aggregation rule	Version	Best (η, α)	RMSE $\times 10^3$	MAE $\times 10^3$	MAPE $\times 10^2$	CORR %
$\mathcal{F}_{\eta\alpha}$	RMSE	$(1400, 0.05)$	632 ± 11	471 ± 7	0.832 ± 0.012	99.8
	MAE	$(100, 0.01)$	636 ± 13	468 ± 7	0.828 ± 0.012	99.8
	MAPE	$(9000, 0.01)$	636 ± 13	468 ± 7	0.828 ± 0.012	99.8
$\mathcal{F}_{\eta\alpha}^{\text{grad}}$	RMSE[a]	$(1, 0.01)$	599 ± 9	450 ± 6	0.798 ± 11	99.9
	MAE	$(0.5, 0.01)$	622 ± 10	464 ± 7	0.820 ± 11	99.8
	MAPE	$(9, 0.01)$	625 ± 10	468 ± 7	0.828 ± 11	99.8

[a]This version reaches very good results. It is however very unstable and sensitive to noise and has basically similar performance as MAE or MAPE version.

Table 5: Performance of $\mathcal{F}_{\eta\alpha}$ and $\mathcal{F}_{\eta\alpha}^{\text{grad}}$ with the best constant parameters on the grid $\Lambda_{\mathcal{F}}$.

Performance of adaptive aggregation rules

In Tables 6, 7, and 8, we report the performance of the aggregation rules when performing an online calibration of the parameters.

We present in Algorithm 2 the method used to extend the EWA and specialist aggregation rules into adaptive algorithms. In the experiments, we started around the theoretical optimal value, $\Lambda = \{1/B\}$. Since the average error is not necessarily a convex function of the parameter, we must deal carefully with the initialization. Too small seed values lead to long convergence time and performance close to the performance of $\mathcal{U}_m$. On the contrary, too large parameters lead to instability and may converge to an optimum which is only locally optimal.

When the optimal parameter $\eta^\star$ reached the lower (resp., upper) bound of Λ, we enlarged the grid by adding the values $\eta^\star/2$, $\eta^\star/4$, and $\eta^\star/8$ (resp., $2\eta^\star$, $4\eta^\star$, and $8\eta^\star$). It is of course not the only way to enlarge the grid Λ but it seems a good trade-off between complexity and performance. Indeed, we tried to increase the grid with more values each time, but the benefit was not considerable. We also tested different values for the increase factor and 2 gave good results.

Algorithm 2 Extension of a family of aggregation rules $(\mathcal{A}_\eta)$ abiding by the operational constraint to an adaptive aggregation rule.

`Input:` initial grid Λ of possible parameters, family of algorithms $(\mathcal{A}_\eta)$ abiding by the operational constraint

`Initialization:` $\eta^\star \leftarrow$ randomly sampled $\in \Lambda$ // see footnote [a]
`for` instance t `from` 1 `to` T `do`
  `predict` $\widehat{y}_t \leftarrow \widehat{y}_t\,(\mathcal{A}_{\eta^\star})$ // see footnote [b]
  `if` $t = 48k$ `for some` k `update`
    $\eta^\star \in \arg\min_{\eta\in\Lambda} \overline{\mathrm{ERR}}_t\,(\mathcal{A}_\eta)$
    `if` $\eta^\star \in \partial\Lambda$ `perform` logarithmic enlargement of Λ // see footnote [c]
  `end if`
`end for`

[a] Remember that the most of the aggregation rules use a uniform average of the predictions of active experts until they get access to a feedback (i.e., until instance $t = 48$ when the operational constraint needs to be abided by).

[b] $\widehat{y}_t(\mathcal{A}_\eta)$ is the prediction of aggregation rule $\mathcal{A}_\eta$ (with constant parameter η) after observing the sequence $y_1, \ldots, y_t$ and the corresponding experts predictions; remember that $\mathcal{A}_\eta$ abides by the operational constraint and thus cannot necessarily use the whole set of past observations.

[c] $\partial\Lambda$ denotes the borders of Λ.

Since it is bounded in $[0, 1]$, we can consider a fixed grid for the mixing rate α of fixed-share type aggregation rule. We take

$$\alpha \in \left\{0\,,\ 0.005\,,\ 0.01\,,\ 0.05\,,\ 0.1\,,\ 0.2\,,\ 0.5\,,\ 1\right\}.$$

The adaptive version of fixed-share type aggregation rules then follows, by considering an adaptive grid for η as we did before and by choosing at each round the best constant couple of parameters (η, α) in hindsight in the current grid.

Aggregation rule	Version	RMSE $\times 10^3$	MAE $\times 10^3$	MAPE $\times 10^2$	CORR %
Family $\mathcal{W}$	RMSE	724 ± 11	553 ± 7	0.979 ± 0.013	99.8
	MAE	728 ± 13	549 ± 8	0.976 ± 0.013	99.8
	MAPE	724 ± 12	545 ± 8	0.969 ± 0.013	99.8
Family $\mathcal{W}^{\text{grad}}$	RMSE	640 ± 9	490 ± 7	0.874 ± 0.011	99.8
	MAE	638 ± 9	487 ± 6	0.868 ± 0.011	99.8
	MAPE	634 ± 9	486 ± 6	0.861 ± 0.011	99.8

Table 6: Performance of the adaptive versions of $\mathcal{W}$ and $\mathcal{W}_{\text{grad}}$.

Aggregation rule	Version	RMSE $\times 10^3$	MAE $\times 10^3$	MAPE $\times 10^2$	CORR %
Family $\mathcal{S}$	RMSE	723 ± 11	552 ± 7	0.977 ± 0.013	99.8
	MAE	726 ± 12	549 ± 8	0.975 ± 0.015	99.8
	MAPE	725 ± 12	547 ± 8	0.970 ± 0.013	99.8
Family $\mathcal{S}^{\text{grad}}$	RMSE	640 ± 9	490 ± 7	0.874 ± 0.011	99.8
	MAE	646 ± 10	491 ± 7	0.874 ± 0.012	99.8
	MAPE	652 ± 10	494 ± 7	0.880 ± 0.012	99.8

Table 7: Performance of the adaptive versions of $\mathcal{S}$ and $\mathcal{S}^{\text{grad}}$.

Aggregation rule	Version	RMSE $\times 10^3$	MAE $\times 10^3$	MAPE $\times 10^2$	CORR %
Family $\mathcal{F}$	RMSE	658 ± 12	488 ± 7	0.865 ± 0.012	99.8
	MAE	664 ± 12	496 ± 7	0.877 ± 0.012	99.8
	MAPE	667 ± 12	493 ± 7	0.869 ± 0.012	99.8
Family $\mathcal{F}^{\text{grad}}$	RMSE	623 ± 11	463 ± 7	0.822 ± 0.012	99.8
	MAE	637 ± 10	477 ± 7	0.844 ± 0.012	99.8
	MAPE	656 ± 12	485 ± 7	0.856 ± 0.012	99.8

Table 8: Performance of the adaptive versions of $\mathcal{F}$ and $\mathcal{F}^{\text{grad}}$.

3. Ridge regression aggregation rule

We consider an additional aggregation rule aiming at coming close to the performance of the best fixed linear expert $\mathcal{O}_{\mathbb{R}^N}$. However, no theoretical bound in the context of specialized experts is associated with it yet. The pseudo-code is given in Algorithm 3.

Algorithm 3 Ridge regression aggregation rule $\mathcal{R}_\lambda$ and $\mathcal{R}_\lambda^{\text{grad}}$

`Input:` regularization parameter λ

`Initialization:` $u_1 = (1/N, \dots, 1/N)$

`for` instance t `from` 1 `to` T `do`

`predict` $\widehat{y}_t \leftarrow \dfrac{1}{\tau_t(\boldsymbol{u}_t)} \displaystyle\sum_{j \in E_t} u_{jt} f_{jt}$

$\boldsymbol{u}_{t+1} \leftarrow \underset{\boldsymbol{u} \in \mathbb{R}^d}{\arg\min} \displaystyle\sum_{s=2}^{T} \ell_s\left(\frac{\boldsymbol{u}}{\tau_s(\boldsymbol{u})}\right) \left|\tau_s(\boldsymbol{u})\right| + \lambda \left\|u\right\|_2^2$ // see footnote [a]

`end for`

[a] $\mathcal{R}_\lambda^{\text{grad}}$ corresponds to the replacement of the loss function ℓ_s by the pseudo-loss function $\tilde{\ell}_s$ defined for all $\boldsymbol{u} \in \mathbb{R}^N$ by $\tilde{\ell}_s(\boldsymbol{u}) = \nabla \ell_s(\boldsymbol{u}_s) \cdot \boldsymbol{u}$, where $\nabla \ell_s$ denotes a subgradient of ℓ_s.

Its performance for the best fixed parameter λ on the grid

$$\Lambda_{\mathcal{R}} = \left\{10^i,\ i \in \{-6, \dots, 7\}\right\}$$

is reported in Table 9.

Aggregation rule	Version	Best λ	RMSE $\times 10^3$	MAE $\times 10^3$	MAPE $\times 10^2$	CORR %
$\mathcal{R}_\lambda$	RMSE	$1 \cdot 10^3$	650 ± 9	502 ± 7	0.898 ± 0.011	99.8
	MAE	$1 \cdot 10^3$	681 ± 9	527 ± 7	0.950 ± 0.013	99.8
	MAPE	$1 \cdot 10^2$	661 ± 10	504 ± 7	0.897 ± 0.012	99.8
$\mathcal{R}_\lambda^{\text{grad}}$	RMSE	Poor throughout this table				
	MAE					
	MAPE					

Table 9: Performance of $\mathcal{R}_\lambda$ with the best constant parameters on the grid $\Lambda_{\mathcal{R}}$.

Ce n'est pas la bonne façon de faire une descente de gradient avec Ridge. Regarder la méthode proposée dans [..., demander à Séb.] ou y réfléchir.

4. Random Forests

We adopt a stochastic approach in this section. The electricity consumption $(Y_t) \in \mathbb{R}^T$ and the expert advice $(F_{jt}) \in \mathbb{R}^{N\times T}$ are now modeled as random processes. They may depend on contextual variables $(X_t) \in \mathbb{R}^{d\times T}$ that can be observed before the prediction is formed. We denote by $(\mathcal{F}_t^{\text{context}})$ and $(\mathcal{F}_t^{\text{exp}})$ two filtrations of the past information, defined for all $t \geqslant 1$ by

$$\begin{aligned}\mathcal{F}_t^{\text{context}} &= \sigma\Big(\big\{(X_s, Y_s)\,,\ 1 \leqslant s \leqslant t-1\big\} \cup \big\{X_t\big\}\Big)\,,\\ \mathcal{F}_t^{\text{exp}} &= \sigma\Big(\big\{(F_{js}, Y_s)\,,\ 1 \leqslant s \leqslant t-1,\ 1 \leqslant j \leqslant N\big\} \cup \big\{F_{jt},\ 1 \leqslant j \leqslant N\big\}\Big)\,.\end{aligned}$$

The expert advice F_{jt} are assumed to be $\mathcal{F}_t^{\text{context}}$–measurable. Hence, $\mathcal{F}_t^{\text{exp}}$ is contained in $\mathcal{F}_t^{\text{context}}$. We furthermore assume that the electricity consumption Y_t can be written as

$$Y_t = \mathbb{E}_t[Y_t] + \varepsilon_t\,,$$

where $\mathbb{E}_t[\,\cdot\,] = \mathbb{E}\left[\,\cdot\,|\mathcal{F}_t^{\text{context}}\right]$ denotes the conditional expectation given the past $\mathcal{F}_t^{\text{context}}$ and each $\varepsilon_t \sim \mathcal{N}\big(0, \sigma^2\big)$ is a Gaussian random noise independent from $\mathcal{F}_t^{\text{context}}$.

In the model above, we can use the random forest regression method for three different goals. First, we aim at building a new stochastic expert that estimates $\mathbb{E}_t\big[Y_t\big]$; we could then add this predictor in our set of experts. Then, we could compare the performance of the aggregation rules of the previous sections with stochastic methods that estimate at each time instance $\mathbb{E}\big[Y_t|\mathcal{F}_t^{\text{exp}}\big]$. Finally, it could be interesting to create a new aggregation rule that takes into account the contextual information (X_t) before proposing a mixture, as the performance of the experts may actually depend on the contextual variables (X_t).

A brief description of random forest regression algorithm

Before we do so, we propose a brief overview of random forests. Random forests have been proposed by Leo Breiman [**?**]. They are implemented in `R` with the package `randomForest`. For more details about the implementation and the use of the package `R`, the reader is referred to [**?**].

We have at our disposal a training data set $(X_t, Y_t)_{t\in S_0}$ of pairs of observed contextual variables X_t and output variables Y_t, where the set S_0 is formed by all training instances. The contextual variables take values in $\mathcal{C}$. They may be multivariate and consist of $M \geqslant 1$ real or categorical features. A random forest is a collection of tree predictors $h_k : \mathcal{C} \to \mathbb{R}$, for $k = 1, \ldots, K$, where $\boldsymbol{x}$ denotes the observed input. The random forest prediction is the unweighted average over the collection,

$$h : \boldsymbol{x} \mapsto \frac{1}{K}\sum_{k=1}^{K} h_k(\boldsymbol{x})\,.$$

Let $T_0 = |S_0|$ be the size of our whole training set. A tree predictor is built in the following way.

We first proceed by bootstrapping; that is, we choose a smaller training set S_k for this particular tree by sampling T_0 times –uniformly at random and with replacement– in S_0. About a third of the available data is hence put aside and can be used as a testing set for this tree.

Then, we let the tree grow, by going down from the root to the leaves as follows. For each node of the tree, we randomly choose $m \simeq M/3$ features on which to base the decision at that node and choose the best split among these features[1]. The tree grows until the number of contextual variables X_t per terminal node is small enough ($\leqslant 5$).

[1]The bagging procedure proposed by Breiman [**?**] is the special case where $m = M$. Its drawback is however to create more correlated trees and to increase the variance of the final prediction.

A given tree finally predicts, for the contextual variable $\boldsymbol{x}$, as the unweighted average over the outputs Y_t of pairs (X_t, Y_t) whose contextual variables belong to the same leaf as $\boldsymbol{x}$.

Random forests define some notion of proximity between the contextual variables. After the trees are grown, we may run the data on them. We then define the proximity between two instances t_1 and t_2 as

$$\mathrm{prox}(t_1, t_2) = \frac{1}{K} \sum_{k=1}^{K} \mathbb{1}\{\text{contextual variables } X_{t_1} \text{ and } X_{t_2} \text{ are in the same leaf in tree } k\} \cdot$$

4.1. Random Forests as a way to construct new experts based on contextual data

We consider here random forests as a way to estimate $\mathbb{E}_t\big[Y_t\big]$ and to produce some new experts (F_{jt}), with $j > N$ and $1 \leqslant t \leqslant T$.

The training set consists of the input–output pairs (X_t, Y_t) from September 1, 2002 to August 31, 2007. The considered contextual variables are calendar variables (type of day –see Table 13–, season, and so on), weather variables (temperature, nebulosity, wind, etc.), and time series variables (consumption of the last day). We refer the interested reader to page 28 for a more detailed description of the data.

We deal independently with each half-hour consumption and hence consider 48 separate time series. The basic random forest predictor is denoted by RF_0. We report in Figure 3 its performance on the prediction set, that is, from September 1, 2007 to August 31, 2008. This predictor is however a bit too unspecific and requires some improvements driven by our data in order to exhibit better performance. We hence built new predictors RF_1–RF_4, which are presented below in an increasing order of sophistication. Their features are summarized in Table 10.

Understand when huge errors occur and correct it

- *Online training set:* This improvement, corresponding to the predictors RF_1–RF_4, consists in taking into account also the data of the prediction set available so far, that is, until the day before. It is motivated by the fact that RF_0 reaches poor results for unexplored area, like at the end of December 2007 when the energy consumption was larger then ever before. E.g., RF_1 predicts the energy consumption at time instance t by

$$\widehat{Y}_t = \frac{1}{\sum_{s \in A_t} \mathrm{prox}(t, s)} \sum_{s \in A_t} \mathrm{prox}(t, s)\, Y_s \,,$$

  where A_t denotes all the time instances corresponding to the same hour as t in the training set and in the prediction set available so far. The rules RF_2–RF_4 are described below.

- *Linear model:* It consists in predicting a linear model of some $m \leqslant M$ contextual variables instead of simply computing a weighted average of the past observations Y_s. This way, the predictor should be able to better extrapolate the information gained so far to new unexplored area. We denote by $\phi : \{1, \ldots, m\} \to \{1, \ldots, M\}$ the injective mapping determining the m attributes considered in the linear model. The predictor then resorts, at time instance t, to

$$\widehat{Y}_t = \sum_{n=1}^{m} X_{\phi(n)t} \widehat{a}_{nt} + \widehat{a}_{m+1t} \,, \qquad \text{(Linear Model)}$$

  where $(\widehat{a}_t) \in \underset{a \in \mathbb{R}^{m+1}}{\arg\min} \sum_{s \in S_t} \mathrm{prox}(t, s) \left(Y_s - \sum_{n=1}^{m} X_{\phi(n)s} a_n - a_{m+1} \right)^2 .$

  We constructed two such predictors. For the first one, RF_2, we took $m = 1$ and considered only the energy consumption before the last time update (i.e., last midday), while for the second one, RF_3, we took $m = 2$ and considered also the temperature. Their results are reported in Figure 3.

- *Bias correction:* since the energy consumption is not really a stationary process but suffers from some drift (towards higher consumptions), the predictions happen to be biased. We may want to adjust them. To do so we may estimate the current bias before the prediction is formed and correct it. Formally, we consider a ridge-like correction:

$$\widehat{Y}'_t = \widehat{\theta}_t \widehat{Y}_t \,, \qquad \text{(Bias correction)}$$

where

$$\widehat{\theta}_t = \operatorname*{arg\,min}_{\theta \in \mathbb{R}} \sum_{s \in S_t} \left(Y_s - \theta \widehat{Y}_s \right)^2 = \frac{\sum_{s \in S_t} X_s Y_s}{\sum_{s \in S_t} X_s^2}$$

is the estimation of the current (multiplicative) bias and $S_t = [\![\sigma(t) - h, \sigma(t)]\!]$ denotes a sliding time window of length h and ending at the last time update $\sigma(t)$ due to the operational constraint. Other estimates of the bias, such as $\widehat{\theta}_t = 1/|S_t| \sum_{s \in S_t} Y_s / X_s$, exhibit a similar performance.

Predictor	Basic Properties	Online training set	Linear Model	Bias correction
RF_0	×			
RF_1	×	×		
RF_2	×	×	×	
RF_3	×	×	×	
RF_4	×	×	×	×

Table 10: Properties of the random forest predictors.

4.2. Random Forests as a way to construct new experts based on expert advice only

We use here the random forests to estimate $\mathbb{E}\big[Y_t | \mathcal{F}_t^{\text{exp}}\big]$ at each time instance; that is, we build some estimate $\widehat{Y}_{t+1}$ from the expert advice F_{js}. The considered process does however not return a convex combination. Its performance can be compared to the performance of the aggregation rules considered in the previous section. It exhibits poor results (RMSE of the order of 900).

4.3. Random Forests as a stochastic aggregation rule

The idea is to consider contextual variables (X_t) available at the beginning of each round and whose values is a good indicator on the expected performance of the different experts.

At instance t, we denote by $\widehat{Y}_t$ the prediction of this aggregation rule and by $\widehat{L}_t = \ell(\widehat{Y}_t, Y_t)$ and $L_{it} = \ell(F_{it}, Y_t)$ the losses respectively suffered by it and by expert i. We assume that the losses L_{it} can be modeled as

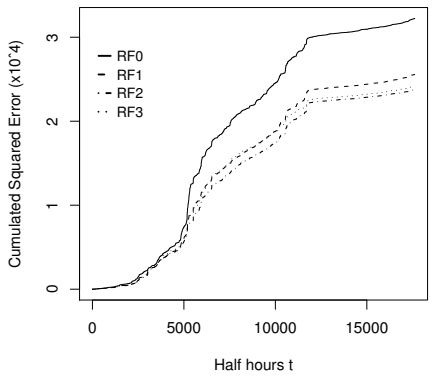
$$L_{it} = \mathbb{E}_t[L_{it}] + \varepsilon'_{it} \,,$$

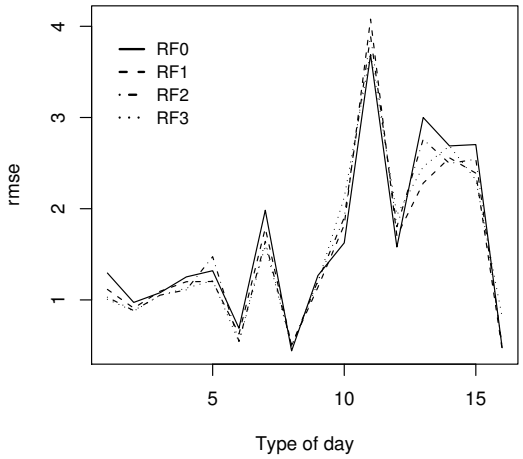
where $\varepsilon'_{it} \sim \mathcal{N}(0, {\sigma'_i}^2)$ is a Gaussian noise independent from the past (i.e., from $\mathcal{F}_t^{\text{context}}$).

At instance t, we get an estimation $\widehat{\ell}_{it}$ of the conditional expectation of the loss suffered by expert i. This estimation can be got thanks to any regression algorithm such as random forests. We suppose that it can be decomposed as

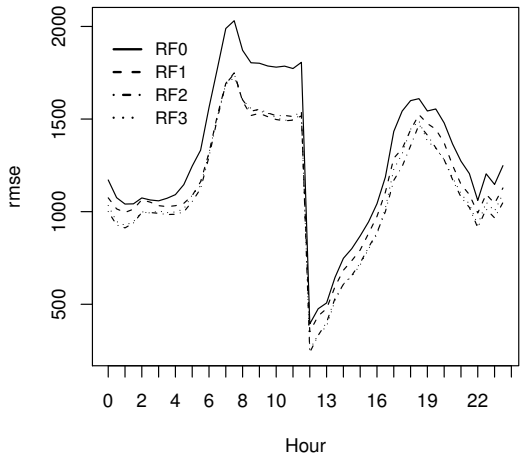
$$\widehat{\ell}_{it} = \mathbb{E}_t[L_{it}] + \varepsilon_{it} \,,$$

where $\varepsilon_{it} \sim \mathcal{N}\big(0, \sigma_t^2\big)$ is a Gaussian random noise independent from the past $\mathcal{F}_t^{\text{context}}$ and from the other noises $(\varepsilon_{jt})_{j \neq i}$ and (ε'_{jt}) .





Modifier rmse vers RMSE (pour mémoire : qd je referai les graphes)



RMSE $\times 10^3$	366 days	320 days	EJP[a]	not EJP
RF_0	1353	1297	2114	1269
RF_1	1204	1133	1690	1155
RF_2	1161	1074	1653	1121
RF_3	1171	1096	1580	1119
RF_4	$\simeq 1130$	A compléter (à EDF)		

[a]EJP stands for "Effacement Jour de Pointe". It is a special fare that discourages some industries to consume electricity; there are 28 such.

Figure 3: Some graphics resuming the behaviour of random forest predictors. For more details on the description of the data, see page 28.

More specifically, in our experiments, we build a basic random forest using the information available so far as a training set $(X_s, Y_s)_{1 \leqslant s \leqslant t-1}$. The estimate of the loss of expert i is then given by

$$\widehat{\ell}_{it} = \frac{1}{\sum_{s=1}^{t-1} \mathrm{prox}(s,t)} \sum_{s=1}^{t-1} \mathrm{prox}(s,t)\, L_{is}\,. \tag{1}$$

We deduce the stochastic aggregation rule stated Algorithm 4.

Trouver une façon de faire une descente de gradient. L'algo est pour l'instant à comparer à EWA, ce serait bien d'en avoir un comparable à EG.

Une meilleure complexité en N ?

Theorem 1. *Under the previous assumptions, the regret of the random forests aggregation rule described above, computed with the sequence of learning rates (η_t), can be bounded by*

$$\sum_{t=1}^{T} \mathbb{E}_t\Big[\widehat{L}_t\Big] - \min_{1 \leqslant j \leqslant N} \mathbb{E}_t[L_{jt}] \leqslant N^2 \left(\sum_{t=1}^{T} \sigma_t + \frac{1}{N\eta_t} \right).$$

Algorithm 4 Random forests aggregation rule (AggRF)

`Input:` sequence of learning rates (η_t), possibly constant

`for` instance t `from` 1 `to` T `do`
  `Construct` the random forest with the data $(X_s, Y_s)_{s \leqslant t-1}$ observed so far
  `for` expert j `from` 1 `to` N `do`
    `observe` F_{jt}
    `get` $\widehat{\ell}_{jt}$ // estimate with the forest constructed above using (1)
    $w_{jt} \leftarrow e^{-\eta_t \widehat{\ell}_{jt}}$ // update
  `end for`
  `predict` $\widehat{y}_t \leftarrow \dfrac{1}{\sum_{i \in E_t} w_{it}} \displaystyle\sum_{j \in E_t} w_{jt} F_{jt}$
  `observe` y_t
`end for`

The best theoretical value is $\eta_t = +\infty$ in view of this bound. It amounts to give a weight of 1 to the best expected expert and 0 to all other experts; that is, to follow the expert with the smallest estimate $\widehat{\ell}_{jt}$ of expected loss. The bound then becomes

$$\sum_{t=1}^{T} \mathbb{E}_t\Big[\widehat{L}_t\Big] - \min_{1 \leqslant j \leqslant N} \mathbb{E}_t[L_{jt}] \leqslant N^2 \sum_{t=1}^{T} \sigma_t$$

(and can easily be improved into a bound linear in N). However, for $\eta_t \simeq (1/N) \sum_{t=1}^{T} \sigma_t$, we loose no more than a constant factor in T with respect to this theoretical optimal bound while the practical performance may be improved as can be seen in Table 12.

The bound is written with a parameter η_t that can evolve with the time t. However, in the experiments, we only use fixed parameters η. In a second step, it would be interesting to study an on-line calibration of the parameter η_t, as we did for the other aggregation rules.

The performance of the random forest aggregation rule –described in Algorithm 4– for various choices of fixed parameter η is summarized in Table 11. The performance obtained for the best choice of η in the grid

$$\Lambda_{RF} = \Big\{ m \cdot 10^k, \quad m \in \big\{1, \ldots, 9\big\} \quad \text{and} \quad k \in \big\{-6, \ldots, \infty\big\} \Big\}$$

is summarized in Table 11.

η	10^{-4}	10^{-3}	10^{-2}	10^{-1}	1	10^{1}	10^{2}	10^{3}	10^{4}
RMSE $\times 10^3$	724	724	721	705	679	692	735	743	743

Table 11: Performance of the random forest aggregation rule, for various choices of constant values of the learning parameter η. The best performance is actually obtained for $\eta = 2$ (value not reported).

Proof. Let $t \geqslant 1$. We define

$$r_t = \mathbb{E}_t\Big[\widehat{L}_t\Big] - \mathbb{E}_t[L_{j^* t}] \ ,$$

best η	RMSE $\times 10^3$	MAE $\times 10^3$	MAPE $\times 10^2$	CORR %
Empirical: 2	676 ± 13	506 ± 7	0.895 ± 0.012	99.8
Theoretical: ∞	743 ± 20	547 ± 8	0.972 ± 0.014	99.8

Table 12: Comparison of the performance of the AggRF with the empirical best constant parameter η on the grid Λ_{RF} (see Table 11) and with the theoretical best η (i.e., $\eta = \infty$).

where $j^* \in \underset{1 \leqslant j \leqslant N}{\arg\min}\, \mathbb{E}_t[L_{jt}]$. We aim at proving that $r_t \leqslant 3N^2\sigma_t/\sqrt{2\pi}$.

By convexity of the loss function ℓ in its first argument, and by conditional independence of (L_{jt}) and $(\widehat{\ell}_{it})$ given $\mathcal{F}_t^{\text{context}}$, we have

$$
\begin{aligned}
r_t &\leqslant \mathbb{E}_t\left[\sum_{j=1}^N \frac{e^{-\eta_t \widehat{\ell}_{jt}}}{\sum_{i=1}^N e^{-\eta_t \widehat{\ell}_{it}}} L_{jt}\right] - \mathbb{E}_t[L_{j^*t}] && \text{// by convexity of } \ell \\
&= \mathbb{E}_t\left[\sum_{j=1}^N \frac{e^{-\eta_t \widehat{\ell}_{jt}}}{\sum_{i=1}^N e^{-\eta_t \widehat{\ell}_{it}}} \mathbb{E}_t[L_{jt}] - \mathbb{E}_t[L_{j^*t}]\right] && \text{// by independence of } (L_{jt}) \text{ and } \left(\widehat{\ell}_{it}\right) \text{ given } \mathcal{F}_t^{\text{context}} \\
&= \mathbb{E}_t\left[\sum_{j=1}^N \frac{\mathbb{E}_t[L_{jt}] - \mathbb{E}_t[L_{j^*t}]}{1 + \sum_{i \neq j} e^{-\eta_t\left(\widehat{\ell}_{it} - \widehat{\ell}_{jt}\right)}}\right] \\
&\leqslant \mathbb{E}_t\left[\sum_{j=1}^N \frac{\mathbb{E}_t[L_{jt}] - \mathbb{E}_t[L_{j^*t}]}{1 + e^{\eta_t\left(\widehat{\ell}_{jt} - \widehat{\ell}_{j^*t}\right)}}\right] && \text{// we lower bounded the denominator}
\end{aligned}
$$

We now use the decomposition of the $\widehat{\ell}_{jt}$ and the fact that by definition, $\mathbb{E}_t[L_{jt}] - \mathbb{E}_t[L_{j^*t}] \geqslant 0$, to get the bound

$$
r_t = \mathbb{E}_t\left[\sum_{j=1}^N \frac{\mathbb{E}_t[L_{jt}] - \mathbb{E}_t[L_{j^*t}]}{1 + e^{\eta_t\left(\mathbb{E}_t[L_{jt}] - \mathbb{E}_t\left[L_{j^*t}\right] + \varepsilon_{jt} - \varepsilon_{j^*t}\right)}}\right] \leqslant N\ \mathbb{E}_t\left[\max_{x \in \mathbb{R}_+} \frac{x}{1 + e^{\eta_t(x - \varepsilon_t'')}}\right],
$$

where $\varepsilon_t'' = \sum_{j=1}^N |\varepsilon_{jt}|$. Since $\eta_t \geqslant 0$, the function

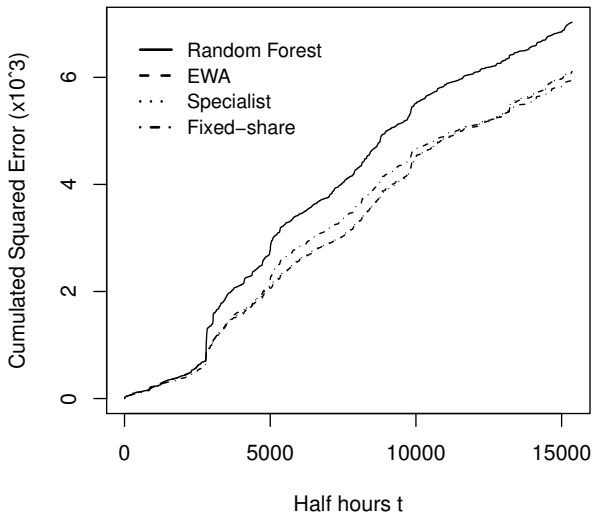
$$
x \longmapsto \frac{x}{1 + e^{\eta_t(x - \varepsilon_t'')}}
$$

is bounded over $\mathbb{R}_+$. Its maximum is given by a well-know function, the Lambert W function (or Omega function), which is the inverse mapping of the function $x \in \mathbb{R}_+ \mapsto xe^x$. Therefore,

$$
r_t \leqslant N\ \mathbb{E}_t\left[\frac{W\left(e^{-1+\eta_t\varepsilon_t''}\right)}{\eta_t}\right].
$$

Now, W is increasing and satisfies the following two inequalities,

$$
\begin{aligned}
\forall x \geqslant 1, \qquad & W(e^x) \leqslant x\,, \\
\forall x \leqslant 1, \qquad & W(e^x) \leqslant W(e) = 1\,,
\end{aligned}
$$



Mixture [a]	RF	$\mathcal{W}^{\text{grad}}$	$\mathcal{S}^{\text{grad}}$	$\mathcal{F}^{\text{grad}}$	$\mathcal{M}$
All days	676	629	631	622	615
EJP	822	723	666	738	666
not EJP	660	619	627	610	610

[a] We used the version MAE of $\mathcal{F}^{\text{grad}}$ because of the instability of the RMSE version. $\mathcal{M}$ is a master algorithm that predicts as $\mathcal{F}^{\text{grad}}$ for normal days and as $\mathcal{S}^{\text{grad}}$ for EJP.

Figure 4: Cumulated squared losses of the different mixture algorithms for their best fixed parameters. We remark that the performance of the specialist and EWA forecasters coincide almost exactly.

so that

$$\begin{aligned}
\mathbb{E}_t\Big[W\Big(e^{-1+\eta_t\varepsilon_t''}\Big)\Big] &\leqslant \mathbb{E}_t\Big[W\Big(e^{-1+\eta_t\varepsilon_t''}\Big)\,\mathbb{1}_{\{\varepsilon_t''\geqslant 2/\eta_t\}} + W\Big(e^{-1+\eta_t\varepsilon_t''}\Big)\,\mathbb{1}_{\{\varepsilon_t''<2/\eta_t\}}\Big] \\
&\leqslant \mathbb{E}_t\Big[\big(-1+\eta_t\varepsilon_t''\big)\,\mathbb{1}_{\{\varepsilon_t''\geqslant 2/\eta_t\}} + \mathbb{1}_{\{\varepsilon_t''<2/\eta_t\}}\Big] \\
&\leqslant \mathbb{E}_t\Big[\eta_t\,\varepsilon_t''\,\mathbb{1}_{\{\varepsilon_t''\geqslant 2/\eta_t\}} + 1\Big] \\
&\leqslant \eta_t\,\mathbb{E}_t\big[\varepsilon_t''\big] + 1\,. \qquad \text{// since } \varepsilon_t'' \geqslant 0
\end{aligned}$$

This leads to the bound

$$r_t \leqslant N\left(\mathbb{E}_t\big[\varepsilon_t''\big] + \frac{1}{\eta_t}\right) = N\left(N\,\mathbb{E}_t\Big[|\varepsilon|\Big] + \frac{1}{\eta_t}\right),$$

where $\varepsilon \sim \mathcal{N}\big(0,\sigma_t^2\big)$ and where the equality holds by definition of ε_t''. We also use that by the Cauchy-Schwarz inequality, $\mathbb{E}_t[|\varepsilon|] \leqslant \sqrt{\mathbb{E}_t[\varepsilon^2]} = \sigma_t$. Hence, $r_t \leqslant N\,(N\sigma_t + 1/\eta_t)$; summing this inequality over t concludes the proof. □

Appendix A

In this appendix, we justify the measure of dispersion considered for the square loss. Let X be a random variable with mean $\mu = \mathbb{E}\big[X\big]$ and standard deviation $\sigma = \sqrt{\mathrm{Var}\big(X\big)}$. Let $(X_1, \ldots, X_T)$ be a sequence of independent random variables identically distributed according to the distribution of X. We aim at giving a measure of dispersion for $\sqrt{\mathbb{E}\big[X\big]}$. According to the central limit theorem, we have

$$\sqrt{T}\left(\frac{\overline{X}_T - \mu}{\sigma}\right) \rightsquigarrow \mathcal{N}\big(0,1\big)\,,$$

as $T \to \infty$, where $\overline{X}_T$ denotes the empirical mean, $\overline{X}_T = \dfrac{1}{T}\displaystyle\sum_{t=1}^{T} X_t\,.$

The delta method then yields

$$\sqrt{T}\left(\sqrt{\overline{X}_T} - \sqrt{\mu}\right) \rightsquigarrow \mathcal{N}\left(0, \left(1/2\sqrt{\mu}\right)^2 \sigma^2\right),$$

which entails,

$$\sqrt{T}\frac{\left(\sqrt{\overline{X}_T} - \sqrt{\mu}\right)}{\sigma/2\sqrt{\mu}} \rightsquigarrow \mathcal{N}\big(0,1\big)\,.$$

Furthermore, by the method of moments, we have

$$\widehat{s}_T \cdot \frac{2\sqrt{\mu}}{\sigma} \xrightarrow{\mathbb{P}} 1\,,$$

where $\widehat{s}_T$ is defined as

$$\widehat{s}_T = \sqrt{\frac{\dfrac{1}{T}\displaystyle\sum_{t=1}^{T}\Big(X_t - \overline{X}_T\Big)^2}{4\,\overline{X}_T}}\,.$$

Hence, Slutsky's lemma yields

$$\sqrt{T}\left(\frac{\sqrt{\overline{X}_T} - \sqrt{\mu}}{\widehat{s}_T}\right) \rightsquigarrow \mathcal{N}(0,1)\ .$$

This leads to an asymptotically $(1-\alpha)$-confidence interval for $\sqrt{\mathbb{E}\big[X\big]}$,

$$I = \left[\sqrt{\overline{X}_T} \pm \frac{z_\alpha}{\sqrt{T}}\widehat{s}_T\right]\,,$$

where z_α denotes the α-quantile of the normal distribution.

Appendix B

We detail here how the performance of the best compound expert with at most m shifts, $\mathcal{O}_\delta^m$, can be obtained by forward dynamic programming. The idea is to update at each time instance t, for each active expert i and each number of shifts m, the cumulative loss $L(m,i,t)$ suffered until round t (round t included) by the best compound expert with at most m shifts and ending with expert i. The final loss of $\mathcal{O}_\delta^m$ will then be given by

$$\overline{\mathrm{ERR}}_T(\mathcal{O}_\delta^m) = \frac{1}{T} \min_{m' \leqslant m} \min_{j \in E_T} L(m', j, T)\,.$$

Algorithm 5 Best compound expert with at most m shifts, $\mathcal{O}_\delta^m$.

`Input:` observations (y_t), expert advice (f_{it}), active sets (E_t), loss function ℓ

`Initialization:` $L(0,i,0) \leftarrow 0$ `for all` $i \in \{1,\dots,N\}$
`for` instance t `from` 1 `to` T `do`
  `for all` number of shifts $m \in \{0,\dots,t-1\}$ `and` experts $i \in E_t$ `update`

$$L(m,i,t) \leftarrow \min\left\{ L(m,i,t-1),\ \min_{j \in E_{t-1} \setminus \{i\}} L(m-1,j,t-1) \right\} + \ell\big(f_{it}, y_t\big)$$

  `end for`
`end for`
`return` $\dfrac{1}{T} \min_{m' \leqslant m} \min_{j \in E_T} L(m', j, T)$

Appendix C

In this appendix, we provide the proof of an improved regret bound in the case of specialized experts. The original proof was proposed by Blum and Mansour [**?**, Section 6]. It relies on an EWA algorithm run with penalized losses (see Algorithm 6). We use the short-hand notation $\ell_{jt} = \ell(f_{jt}, y_t)$ to denote the loss suffered by expert j at time instance $t \geqslant 1$. We furthermore assume that the loss function ℓ is convex in its first argument and has the interval $[0,1]$ as its range. It is important that ℓ takes its values in $[0,1]$ because of the inequalities (5) and (6). When ℓ has values in a different interval $[m, M]$, the values of m and M need to be known beforehand so that the normalized loss function $(\ell - m)/(M - m)$ can be considered instead of ℓ.

Algorithm 6 The Blum-Mansour improved EWA forecaster for specialized exerts.

`Initialization:` uniform weight vector $\boldsymbol{p}_1 \in \mathcal{X}$
`for` instance t `from` 1 `to` T `do`
  `predict` $\widehat{y}_t \leftarrow \sum_{j \in E_t} p_{jt} f_{jt}$
  `incur loss` $\widehat{\ell}_t \leftarrow \ell(\widehat{y}_t, y_t)$
  `for` expert i `from` 1 `to` N `update`
    $w_{it} \leftarrow e^{-\eta_i \sum_{s=1}^{t} (\ell_{is} - e^{-\eta_i} \widehat{\ell}_s) \mathbb{1}_{\{i \in E_s\}}}$
    $p_{it+1} \leftarrow \dfrac{w_{it} \left(1 - e^{-\eta_i}\right) \mathbb{1}_{\{i \in E_{t+1}\}}}{\sum_{j \in E_{t+1}} w_{jt} \left(1 - e^{-\eta_j}\right) \mathbb{1}_{\{j \in E_{t+1}\}}}$
  `end for`
`end for`

Theorem 2. *If the loss function ℓ is convex in its first argument and has range in $[0,1]$, the regret of the Blum and Mansour aggregation rule described above can be bounded by*

$$\sum_{t=1}^{T} \left(\widehat{\ell}_t - \ell_{it} \right) \mathbb{1}_{\{i \in E_t\}} \leqslant \frac{\ln N}{\eta_i} + \eta_i \sum_{t=1}^{T} \mathbb{1}_{\{i \in E_t\}} .$$

Corollary 3. *The choice of $\eta_i = \sqrt{\dfrac{\ln N}{\sum_{t=1}^{T} \mathbb{1}_{\{i \in E_t\}}}}$ in Theorem 2 yields the bound*

$$\forall i \in \{1, \ldots, N\}, \qquad \sum_{t=1}^{T} \left(\widehat{\ell}_t - \ell_{it} \right) \mathbb{1}_{\{i \in E_t\}} \leqslant 2 \sqrt{\ln N \sum_{t=1}^{T} \mathbb{1}_{\{i \in E_t\}}} .$$

Proof. The main idea of the proof is to control $w_{1t} + \cdots + w_{Nt}$; in particular, we will prove by induction that for all $t \geqslant 0$,

$$\sum_{i=1}^{N} w_{it} \leqslant N . \tag{2}$$

This will be enough to draw the conclusion. Indeed, we will get in particular that $w_{iT} \leqslant N$ for all experts i; or, put differently,

$$-\eta_i \sum_{t=1}^{T} \left(\ell_{it} - e^{-\eta_i} \widehat{\ell}_t \right) \mathbb{1}_{\{i \in E_t\}} \leqslant \ln N .$$

Then, since $e^{-x} \geqslant 1 - x$, for all $x \in \mathbb{R}$, we have

$$\sum_{t=1}^{T} \left((1 - \eta_i) \widehat{\ell}_t - \ell_{it} \right) \mathbb{1}_{\{i \in E_t\}} \leqslant \frac{\ln N}{\eta_i} ,$$

$$\sum_{t=1}^{T}\Big(\widehat{\ell}_t-\ell_{it}\Big)\,\mathbb{1}_{\{i\in E_t\}}\leqslant\frac{\ln N}{\eta_i}+\eta_i\sum_{t=1}^{T}\widehat{\ell}_t\,\mathbb{1}_{\{i\in E_t\}}\,,$$

which yields the stated regret bound by bounding the $\widehat{\ell}_t$ by 1.

It therefore only remains to prove (2), which we do by induction on t. By definition of the weights, we have

$$\begin{aligned}\sum_{i=1}^{N}w_{it} &= \sum_{i=1}^{N}w_{it-1}\,\exp\Big(-\eta_i\Big(\ell_{it}-e^{-\eta_i}\widehat{\ell}_t\Big)\mathbb{1}_{\{i\in E_t\}}\Big) &(3)\\ &= \sum_{i=1}^{N}w_{it-1}\,\exp\Big(-\eta_i\ell_{it}\mathbb{1}_{\{i\in E_t\}}\Big)\exp\Big(\eta_i e^{-\eta_i}\widehat{\ell}_t\mathbb{1}_{\{i\in E_t\}}\Big)\,. &(4)\end{aligned}$$

By the convexity of $x\mapsto e^{\eta x}$ in $[0,1]$, for all $\eta\in\mathbb{R}$, we get for all $x\in[0,1]$,

$$e^{\eta x}\leqslant(1-x)\,e^0+xe^{\eta}=1-x\left(1-e^{\eta}\right).$$

This leads to the following two inequalities, that are valid for all $x\in[0,1]$ and $\eta>0$,

$$\begin{aligned}e^{-\eta x} &\leqslant 1-\left(1-e^{-\eta}\right)x\,, &(5)\\ e^{\eta x} &\leqslant 1-\left(1-e^{-\eta}\right)e^{\eta}x\,. &(6)\end{aligned}$$

Hence, (4) entails

$$\begin{aligned}\sum_{i=1}^{N}w_{it} &\leqslant \sum_{i=1}^{N}w_{it-1}\Big(1-\big(1-e^{-\eta_i}\big)\ell_{it}\mathbb{1}_{\{i\in E_t\}}\Big)\Big(1+\big(1-e^{-\eta_i}\big)e^{\eta_i}\,e^{-\eta_i}\widehat{\ell}_t\mathbb{1}_{\{i\in E_t\}}\Big)\\ &\leqslant \sum_{i=1}^{N}w_{it-1}\Big(1+\big(1-e^{-\eta_i}\big)\big(\widehat{\ell}_t-\ell_{it}\big)\mathbb{1}_{\{i\in E_t\}}\Big)\,.\end{aligned}$$

The convexity of the loss function yields $\widehat{\ell}_t\leqslant\sum_{j=1}^{N}p_{jt}\ell_{jt}$, which entails

$$\begin{aligned}\sum_{i=1}^{N}w_{it}-\sum_{i=1}^{N}w_{it-1} &\leqslant \sum_{i=1}^{N}\underbrace{w_{it-1}\big(1-e^{-\eta_i}\big)\mathbb{1}_{\{i\in E_t\}}}_{p_{it}Z_t}\Big(\widehat{\ell}_t-\ell_{it}\Big)\\ &\leqslant \sum_{i=1}^{N}p_{it}Z_t\left(\sum_{j=1}^{N}p_{jt}\ell_{jt}-\ell_{it}\right)\\ &= Z_t\sum_{i=1}^{N}p_{it}\sum_{j=1}^{N}p_{jt}\ell_{jt}-Z_t\sum_{i=1}^{N}p_{it}\ell_{it}=0\,,\end{aligned}$$

where $Z_t=\sum_{j\in E_t}w_{jt-1}\left(1-e^{-\eta_j}\right)\mathbb{1}_{\{j\in E_t\}}$ is the normalization factor in the definition of $\boldsymbol{p}_t$.

This leads to (2) and hence concludes the proof. □

Appendix D

We consider the same setting and notation as in Appendix C. We bound the cumulated loss of an aggregation rule with respect to a given expert j by the variation of the observed losses suffered by this expert. The proof was proposed by Hazan and Kale [**?**]. The considered aggregation rule is a version of EWA computed with penalized losses. At time instance t, the weight of expert i is updated as

$$p_{it+1} \leftarrow \frac{\exp\left(-\eta \sum_{s=1}^{t} \tilde{\ell}_{is}\right)}{W_t} ,$$

where $W_t = \sum_{j=1}^{N} \exp\left(-\eta \sum_{s=1}^{t} \tilde{\ell}_{js}\right)$ is the normalization factor and $\tilde{\ell}_{it} = \ell_{it} + 2\eta\ell_{it}^2$, is the penalized instantaneous loss suffered by expert i at round t.

Theorem 4. *For all $\eta > 0$, if the loss function ℓ is convex in its first argument and is such that $0 \leqslant 2\eta \max_{i,t} \ell_{it} \leqslant 1$, the regret of EWA run over the penalized losses $\tilde{\ell_{it}} = \ell_{it} + 2\eta\ell_{it}$ can be bounded for all experts $j \in \{1, \ldots, N\}$ by*

$$\sum_{t=1}^{T} \widehat{\ell}_t - \sum_{t=1}^{T} \ell_{jt} \leqslant \frac{\ln N}{\eta} + 2\eta\ell_{jt}^2 \; ;$$

that is,

$$\sum_{t=1}^{T} \widehat{\ell}_t \leqslant \frac{\ln N}{\eta} + \min_{j \in \{1,\ldots,N\}} \left\{ \sum_{t=1}^{T} \ell_{jt} + 2\eta\ell_{jt}^2 \right\} .$$

Proof. With the convention that an empty sum is null, we remark that $W_0 = N$. We now lower and upper bound $\ln\left(W_T/W_0\right)$. First,

$$\ln \frac{W_T}{W_0} \;\geqslant\; \ln \frac{\exp\left(-\eta \sum_{t=1}^{T} \tilde{\ell}_{jt}\right)}{N} \;=\; -\eta \min_j \sum_{t=1}^{T} \tilde{\ell}_{jt} - \ln N \, . \tag{7}$$

Second, using that $e^{-x} \leqslant 1 - x + x^2/2$ for all $x \geqslant 0$, we see that for all $1 \leqslant t \leqslant T$,

$$\begin{aligned} \ln \frac{W_t}{W_{t-1}} \;=\; \ln \sum_{j=1}^{N} p_{jt} e^{-\eta\tilde{\ell}_{jt}} \;&\leqslant\; \ln \left(\sum_{j=1}^{N} p_{jt} \left(1 - \eta\tilde{\ell}_{jt} + \frac{\eta^2}{2} \tilde{\ell}_{jt}^2 \right) \right) \\ &\leqslant\; -\eta \sum_{j=1}^{N} p_{jt} \tilde{\ell}_{jt} + \frac{\eta^2}{2} \sum_{j=1}^{N} p_{jt} \tilde{\ell}_{jt}^2 \, . \end{aligned}$$

Hence, by summing over t, we get

$$\ln \frac{W_T}{W_0} \leqslant -\eta \sum_{t=1}^{T} \sum_{j=1}^{N} p_{jt} \tilde{\ell}_{jt} + \frac{\eta^2}{2} \sum_{j=1}^{N} p_{jt} \tilde{\ell}_{jt}^2 \, . \tag{8}$$

Combining (7) and (8) then leads to [2]

$$\sum_{t=1}^{T} \sum_{j=1}^{N} p_{jt} \tilde{\ell}_{jt} \leqslant \frac{\ln N}{\eta} + \frac{\eta}{2} \sum_{j=1}^{N} p_{jt} \tilde{\ell}_{jt}^2 + \min_j \sum_{t=1}^{T} \tilde{\ell}_{jt} \, .$$

[2] This bound is generic and is called a second-order bound on the regret.

We now show that the above second-order bound on penalized losses entails the desired result. Indeed, it can be rewritten as

$$\sum_{t=1}^{T}\sum_{j=1}^{N} p_{jt}\left(\ell_{jt}+2\eta\ell_{jt}^2\right) \leqslant \frac{\ln N}{\eta} + \frac{\eta}{2}\sum_{j=1}^{N} p_{jt}\left(\ell_{jt}+2\eta\ell_{jt}^2\right)^2 + \min_j \sum_{t=1}^{T}\tilde{\ell}_{jt}\,.$$

Now, since $2\eta\ell_{jt} \leqslant 1$ and since the losses are nonnegative, we remark that

$$\frac{\eta}{2}\left(\ell_{jt}+2\eta\ell_{jt}^2\right)^2 \leqslant \frac{\eta}{2}\left(\ell_{jt}+\ell_{jt}\right)^2 = 2\eta\ell_{jt}^2\,.$$

Substituting this bound, we end up with

$$\sum_{t=1}^{T}\sum_{j=1}^{N} p_{jt}\ell_{jt} \leqslant \frac{\ln N}{\eta} + \min_j \sum_{t=1}^{T}\tilde{\ell}_{jt}\,,$$

which, by convexity of the loss in its first argument, concludes the proof.

□

Appendix E

Description of the calendar data used in the random forest section.

`Offset`	
0	Winter hour
1	Summer hour, spring period
10	Summer hour, autumn period
$2, 3, 4$	Winter break
$5, 6, 7, 8, 9$	Summer break

`DayType`	
0	Monday
1	Tuesday, Wednesday, Thursday
2	Friday
3	Saturday
4	Sunday
5	Summer Sunday
6	August Sunday
7	December Sunday
8	Transition between a Sunday and a Monday bank holiday
9	Bank holidays
10	Bridge days
$11, 12, 13$	Christmas ± 1
$14, 15, 16$	New Year ± 1
17	Saturday or bank holidays $+1$

Table 13: Description of the calendar contextual variables `Offset` and `DayType`.

Appendix F – Random Forests

Introduction

Les random forests sont une modification du bagging [3] qui construit une grande collection d'arbres de prédiction décorrélés, avant de les moyenner. Les arbres sont construits profondément, ils ont donc tendance à surapprendre les données. Ils ont un biais faible mais une forte variance. Les moyenner permet, s'ils sont non corrélés, de diminuer la variance, sans augmenter le biais.

Donnons l'intuition. Si les prédictions des arbres sont identiquement distribuées, de variance σ^2, avec un coefficient de corrélation deux à deux ρ, la variance de la moyenne arithmétique de K prédictions est alors,

$$\bar{\sigma}^2 = \frac{1}{K^2}\left(K\sigma^2 + K(K-1)\rho\sigma^2\right) = \rho\sigma^2 + \frac{1-\rho}{K}\sigma^2\,.$$

Quand le nombre K d'arbres dans la forêt augmente, le second terme disparaît mais le premier reste. La corrélation ρ entre les arbres, si elle est trop forte, limite l'intérêt de moyenner. L'idée des random forests est de diminuer la variance $\bar{\sigma}^2$, en décorrélant autant que possible les arbres les uns des autres sans trop augmenter leur variance σ^2. Cela se fait, à l'aide de sélections aléatoires des covariables sur lesquelles on coupe au niveau des nœuds des arbres.

Les arbres construits sont très similaires aux arbres CART. On remarque néanmoins deux légères différences. Dans CART, au niveau d'un nœud, on sélectionne la meilleure coupe parmi toutes les variables contextuelles, et non parmi un sous-ensemble aléatoire. Cet ajout n'a été motivé que par le fait que la forêt contienne plusieurs arbres que l'on veut les plus indépendants possible les uns des autres. Dans les random forests, il n'y a de plus pas d'étape d'élagage (pruning) après la construction de l'arbre, comme dans CART. Cela avait en effet pour fonction de diminuer la variance, ce que l'on fait déjà ici par le bagging.

Références bibliographiques

La méthode a été introduite par Leo Breiman [**?**] bien que de nombreuses idées soient apparues plus tôt dans la littérature, comme le bagging [**?**], ou les arbres CART. Elle est implémentée en `R` avec le paquet `randomForest`, maintenu par Andy Liaw, disponible sur le site du `CRAN`. Le site web `http://www.stat.berkeley.edu/~breiman/RandomForests` donne accès à de la documentation, du code et de nombreux rapport techniques. Une explication détaillée dans le cadre des algorithmes de classification ou de régression est disponible dans le livre [**?**]. Les preuves de convergence ne sont pas évidentes et sont souvent obtenues pour des modèles simplifiés et éloignés de ce qui est considéré en pratique. Dans le cadre de la régression, [**?**] a tenté d'expliquer les random forests par leur similitudes avec les K plus proches voisins. Plus récemment, [**?**] ont prouvé des théorèmes de convergence universelle pour les algorithmes de moyennage, dont les random forests sont un cas particulier.

Le cadre théorique

On dispose d'un ensemble d'entraînement $(X_t, Y_t)_{t\in S_0}$, qui est modélisé par un processus supposé indépendant et identiquement distribué de loi $\mathcal{P}$. Pour $t \geqslant 0$, $X_t = (X_{t1}, \ldots, X_{tM})$ est la réalisation au temps t des M variables contextuelles, observables, pouvant expliquer la sortie $Y_t \in \mathbb{R}$. Chaque

[3] Abbréviation de "bootstrapping and averaging". Méthode qui consiste à construire une collection de prédicteurs faibles en ne sélectionnant qu'une partie de l'ensemble d'entraînement pour chacun d'eux (bootstrapping) avant de les moyenner (averaging).

covariable X_{tm}, $1 \leqslant m \leqslant M$, est à valeurs dans un ensemble $\mathcal{X}_m$, qui est soit muni d'un ordre total, comme $\mathbb{R}$ par exemple, soit fini (ensemble de catégories).

L'objectif est, à partir de l'observation des variables contextuelles X, de prévoir la sortie Y d'un nouveau couple (X, Y) tiré selon $\mathcal{P}$, en faisant la plus petite erreur quadratique possible. Les random forests proposent une solution efficace à ce problème, présenté comme l'algorithme 7.

Algorithm 7 Régression par Random Forest

`Entrées :` $(X_t, Y_t)_{t\in S_0}$, ensemble d'entraînement ; $\boldsymbol{x} \in \mathcal{C}$, covariables pour la valeur à prévoir ; K, nombre d'arbres dans la forêt ; m, nombre de covariables sélectionnées à chaque coupe ; n_{feuille}, nombre de covariables X_t de l'ensemble d'entraînement maximal par feuille.

`Pour` k `de 1 à` K, construire l'arbre T_k ainsi :

1. Choisir un ensemble d'entraînement pour cet arbre (bootstrapping) :
   $S_k \leftarrow$ Tirer N fois avec remise et uniformément dans S_0
2. `Initialiser :` $T_k \leftarrow$ racine contenant tout l'ensemble bootstrap S_0
3. `Tant que` T_k contient une feuille ayant plus de n_{feuille} données `faire`
   a. $M' \leftarrow$ choisir uniformément m parmi les M covariables
   b. Choisir la meilleure variable et la meilleure coupe parmi les m covariables
   $$(j^\star, c^\star) = \operatorname*{arg\,min}_{j\in M',\, c\in C_j} \min_{a\in\mathbb{R}^2} \sum_{l=1}^{2} \sum_{X_i\in c_l} (Y_i - a_l)^2 \,,$$
   où C_j est l'ensemble des façons de couper les données présentes en deux ensembles ordonnés si un ordre est disponible pour j.
   c. Transformer la feuille en un nœud avec deux feuilles. Associer à chaque feuille respectivement les ensembles de variables contextuelles $c_1^\star$ et $c_2^\star$ et les données d'entraînement correspondantes.
4. Faire descendre $\boldsymbol{x}$ dans l'arbre jusqu'à une feuille d'ensemble de covariable associé $\mathcal{F}_k(\boldsymbol{x})$ et prévoir
   $h_k(\boldsymbol{x}) \leftarrow \frac{1}{|\mathcal{F}_k(\boldsymbol{x})|} \sum_{t:X_t\in\mathcal{F}_k(\boldsymbol{x})} Y_t$

`Renvoyer :` $h(\boldsymbol{x}) = \frac{1}{K}\sum_{k=1}^{K} h_k(\boldsymbol{x})$

Trouver la bonne coupe

Chaque nœud interne d'un arbre des forêts aléatoires divise l'ensemble des données en deux. Un point essentiel lors de la construction des arbres est de déterminer la bonne partition qui regroupe au mieux les données selon la variable à expliquer Y_t. Pour diminuer la complexité du problème, on ne considère que les coupes se faisant le long d'une seule variable contextuelle. Notons C_j, l'ensemble des partitions possibles selon la covariable j. Si celle-ci est réelle, on ne considère dans C_j que les partitions en deux intervalles.

On peut définir l'erreur quadratique d'une couple covariable–partition $(j, c) \in \{1, \ldots, M\} \times C_j$, où $c = (c_1, c_2)$ est une partition de l'ensemble $\mathcal{X}_m$ des covariables en deux, par

$$\mathcal{E}(j, c) = \min_{a\in\mathbb{R}^2} \sum_{l=1}^{2} \sum_{X_i\in c_l} (Y_i - a_l)^2 \,.$$

Le meilleur couple covariable–partition d'un sous ensemble de variables contextuelles M', est alors donné par le couple minimisant son erreur quadratique,

$$(j^\star, c^\star) = \operatorname*{arg\,min}_{j\in M',\ c\in C_j} \mathcal{E}(j, c) \,.$$

Pour chaque variable contextuelle, à valeurs dans un ensemble ordonné, la détermination d'une meilleure coupe se fait très rapidement (sa complexité est linéaire en le nombre de données présentes) grâce à l'égalité

$$\arg\min_{a\in\mathbb{R}} \sum_{X_i\in I} (Y_i - a)^2 = \left\{ \frac{1}{\#\{i, X_i \in I\}} \sum_{X_i\in I} Y_i \right\} .$$

Cependant, si la covariable j prend ses valeurs dans un ensemble de q catégories non ordonnées, il y a $2^{q-1}-1$ partitions possibles de l'espace en deux groupes, et le calcul devient vite impossible quand q est grand. La méthode consiste à ordonner les catégories par la moyenne des sorties Y_i appartenant à cette catégorie. On coupe alors comme si on avait une variable ordonnée. On est ramené à une complexité linéaire en q ! On peut montrer que cela donne bien la coupe optimale [**?**]. Bien qu'intuitive la preuve est loin d'être triviale.

Proximités

Les random forests induisent une notion de proximité entre les entrées X_t. C'est l'un des outils les plus utiles de la méthode. Intuitivement, si deux ensembles de covariables X_{t_1} et X_{t_2} tombent souvent dans les mêmes feuilles des arbres, on peut supposer qu'elles expliquent la sortie Y de façon similaire. Plus formellement, après que les arbres de la forêt ont été construits, on fait descendre toutes les données au niveau des feuilles et on définit la proximité entre deux observations t_1 et t_1 par

$$\mathrm{prox}(X_{t_1}, X_{t_2}) = \frac{1}{K} \sum_{k=1}^{K} \mathbb{1}_{\{X_{t_1} \text{ et } X_{t_2} \text{ tombent dans la même feuille dans l'arbre } k\}} .$$

On remarque que la prévision des random forests,

$$h_K(\boldsymbol{x}) = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{\#\mathcal{F}_k(\boldsymbol{x})} \sum_{t:X_t\in\mathcal{F}_k(\boldsymbol{x})} Y_t \,,$$

est proche de

$$\begin{aligned} \frac{1}{\sum_{k=1}^{K} \#\mathcal{F}_k(\boldsymbol{x})} \sum_{k=1}^{K} \sum_{t:X_t\in\mathcal{F}_k(\boldsymbol{x})} Y_t &= \frac{1}{\sum_{t\in S_0} \mathrm{prox}(\boldsymbol{x}, X_t)} \sum_{t\in S_0} \mathrm{prox}(\boldsymbol{x}, X_t)\, Y_t \\ &\in \arg\min_{a\in\mathbb{R}} \sum_{s\in S_t} \mathrm{prox}(t,s)\,(Y_s - a)^2 \,, \end{aligned}$$

où l'égalité procède d'une simple récriture utilisant la définition de la proximité et où la réécriture comme un argmin est obtenue par une minimisation de la perte carrée pondérée par la proximité. C'est cette idée, que nous avons suivi, pour construire le prédicteur RF_1 et ses dérivés de la partie 4.1.

En pratique

Dans mes implémentations des random forests, j'ai pris :

$$\left\{ \begin{array}{rcl} n_{\text{feuille}} &=& 3 \\ m &=& \lfloor M/3 \rfloor \simeq 19 \\ K &=& 400 \end{array} \right.$$

Ridge par FTL

Notations : $\Phi_{t-1}(u) = \sum_{s=1}^{t-1} \underbrace{(u^T x_s - y_s)^2}_{\ell_s(u)} + \lambda \|u\|_2^2$

$$\boxed{u_t = \operatorname{argmin}_{u \in \mathbb{R}^d} \Phi_{t-1}(u)}$$

On veut majorer le regret : $R_n(u) = \sum_{t=1}^{n} \ell_t(u_t) - \sum_{t=1}^{n} \ell_t(u)$

Idée : vol de fuite $\Rightarrow$ regret faible.

Lemme : $\sum_{t=1}^{n} \ell_t(u_{t+1}) - \sum_{t=1}^{n} \ell_t(u) \leq \lambda \left(\|u\|^2 - \|u_1\|^2 \right)$

Preuve : Comme u_{t+1} minimise Φ_t, on a :

$$\begin{aligned} \sum_{s=1}^{t} \ell_s(u) + \lambda \|u\|^2 &\geq \sum_{s=1}^{t} \ell_s(u_{t+1}) + \lambda \|u_{t+1}\|^2 \\ &\geq \ell_t(u_{t+1}) + \Phi_{t-1}(u_{t+1}) \\ &\geq \sum_{s=1}^{t} \ell_s(u_{s+1}) + \lambda \|u_1\|^2 \end{aligned}$$

□

On en déduit que si u_{t+1} et u_t sont proches, le regret est faible.

Corollaire : $\sum_{t=1}^{n} \ell_t(u_t) - \ell_t(u) \leq \sum_{t=1}^{n} \left[\ell_t(u_t) - \ell_t(u_{t+1}) \right] + \lambda \left(\|u\|^2 - \|u_1\|^2 \right)$

$\rightarrow$ On cherche donc à maîtriser $\ell_t(u_t) - \ell_t(u_{t+1})$

Lemme : $\boxed{u_{t+1} = u_t - A_t^{-1}\left(u_t^T x_t - y_t\right) x_t}$ (update)

Preuve : $\nabla \Phi_{t-1}(u_t) = 0$ donc $\sum_{s=1}^{t-1} (u^T x_s - y_s) x_s + \lambda u_t = 0$

donc $-\sum_{s=1}^{t-1} y_s x_s + \left(\lambda I + \sum_{s=1}^{t-1} x_s x_s^T\right) u_t = 0$

donc $A_{t-1} u_t = \sum_{s=1}^{t-1} y_s x_s$ où $\boxed{A_{t-1} = \lambda I + \sum_{s=1}^{t-1} x_s x_s^T}$

On en déduit : $\left(A_t - x_t x_t^T\right) u_t = A_t u_{t+1} - y_t x_t$

comme $x_t^T u_t$ est un scalaire, il vaut sa pauvre $u_t^T x_t$ et on peut le balader □

Lemme : $\boxed{\ell_t(u_t) - \ell_t(u_{t+1}) \leq \ell_t(u_t) \cdot x_t A_t^{-1} x_t}$

Preuve : $\ell_t(u_t) - \ell_t(u_{t+1}) \leq \nabla \ell_t(u_t) \cdot (u_t - u_{t+1})$ par convexité.

$= (u_t^T x_t - y_t) x_t A_t^{-1} (u_t^T x_t - y_t) x_t$ □

Théorème : $\boxed{R_n(u) \leq \lambda\left(\|u\|^2 - \|u_1\|^2\right) + \left(\sum_{i=1}^{d} \ln(\lambda + \lambda_i) - d \ln \lambda\right) \max_{t=1..n} \ell_t(u_t)}$

Preuve : $\sum_{t=1}^{n} x_t^T A_t^{-1} x_t = \sum_{t=1}^{n} \left(1 - \frac{\det A_{t-1}}{\det A_t}\right)$ (Lemme M.11 PLG)

$\leq \sum_{t=1}^{n} \ln \frac{\det A_t}{\det A_{t-1}}$ ($1 - x \leq -\ln x \ \forall x > 0$)

$= \ln \frac{\det A_n}{\det A_0}$

$= \sum_{i=1}^{d} \ln(\lambda + \lambda_i) - d \ln \lambda$

$\begin{cases} \det A_0 = \det(\lambda I) = \lambda^d \\ \det A_n = \prod(\lambda + \lambda_i) \\ \text{où } \lambda_i \text{ vp de } \sum_{t=1}^{n} x_t x_t^T \end{cases}$

□

→ Trade-off λ à optimiser ?