A Further Look at Sequential Aggregation Rules for Ozone Ensemble Forecasting

Vivien Mallet[*†], Sébastien Gerchinovitz[‡*], and Gilles Stoltz[‡§]

History:
This version (version 1) – September 2008

[*]INRIA, Paris-Rocquencourt Research Center, France
[†]Université Paris-Est, CEREA (joint laboratory ENPC - EDF R&D), Marne la Vallée, France
[‡]DMA, École Normale Supérieure, Paris, France
[§]HEC Paris, Jouy-en-Josas, France

Contents

Notations, goals, sample description

Notations

Time rounds $t = 1, 2, \ldots, T$. Stations $\mathcal{N} = \{1, \ldots, S\}$. Base forecasters indexed by $m = 1, \ldots, N$.

The active stations at round t are given by a subset $\mathcal{N}_t \subset \mathcal{N}$. For a round t and $s \in \mathcal{N}_t$, the observation $y_t^s \in [0, B]$ is available (B is a bound on the peak heights).

The prediction of base forecaster m at round t for station s is $x_{m,t}^s$. The vector of predictions for station s at round t is $\mathbf{x}_t^s = (x_{1,t}^s, \ldots, x_{N,t}^s)$.

The master forecaster outputs at each round a vector $\mathbf{v}_t = (v_{1,t}, \ldots, v_{N,t}) \in \mathbb{R}^N$ and forms the prediction, at each station s,

$$\widehat{y}_t^s = \mathbf{v}_t \cdot \mathbf{x}_t^s = \sum_{m=1,\ldots,N} v_{m,t}\, x_{m,t}^s$$

(which has to be compared to y_t^s, if $s \in \mathcal{N}_t$). Two prototypical cases arise,

- we constrain the $\mathbf{v}_t$ to lie in the simplex $\mathcal{X}$ of probability distributions over N elements – in that case, we use the notation $\mathbf{v}_t = \mathbf{p}_t$;
- unconstrained prediction (possibly any vector of $\mathbb{R}^N$) is denoted by $\mathbf{v}_t = \mathbf{u}_t$ in the sequel.

Assessment of the quality of a master forecaster

We use a root mean squared error criterion (RMSE). We start the evaluation from time round $t_0 = 1 + T_0$ (where T_0 usually equals 30, thus allowing for a short learning period). Formally, the RMSE of a sequence $\mathbf{v}_1^T = (\mathbf{v}_1, \ldots, \mathbf{v}_T)$ of prediction vectors is

$$\mathrm{RMSE}_{t_0}(\mathbf{v}_1^T) = \sqrt{\frac{1}{\sum_{t=t_0}^T |\mathcal{N}_t|} \sum_{t=t_0}^T \sum_{s \in \mathcal{N}_t} (\mathbf{v}_t \cdot \mathbf{x}_t^s - y_t^s)^2}\,.$$

Some lower bounds beyond which it is impossible or difficult to go,

- the oracle lower bound, that no forecaster, even knowing the observations beforehand, can beat,

$$B_{\text{oracle}} = \inf_{\mathbf{v}_1,\ldots,\mathbf{v}_T \in \mathbb{R}^N} \mathrm{RMSE}_{t_0}(\mathbf{v}_1^T) = \sqrt{\frac{1}{\sum_{t=t_0}^T |\mathcal{N}_t|} \sum_{t=t_0}^T \inf_{\mathbf{v}_t \in \mathbb{R}^N} \sum_{s \in \mathcal{N}_t} (\mathbf{v}_t \cdot \mathbf{x}_t^s - y_t^s)^2}\,;$$

- the lower bounds on the quantities appearing below in the definition of regret, either

$$B_{\text{regret}_{\mathbb{R}^N}} = \inf_{\mathbf{u} \in \mathbb{R}^N} \mathrm{RMSE}_{t_0}((\mathbf{u}, \ldots, \mathbf{u})) = \inf_{\mathbf{u} \in \mathbb{R}^N} \sqrt{\frac{1}{\sum_{t=t_0}^T |\mathcal{N}_t|} \sum_{t=t_0}^T \sum_{s \in \mathcal{N}_t} (\mathbf{u} \cdot \mathbf{x}_t^s - y_t^s)^2}$$

or

$$B_{\text{regret}\mathcal{X}} = \min_{\mathbf{p}\in\mathcal{X}} \text{RMSE}_{t_0}((\mathbf{p},\dots,\mathbf{p})) = \min_{\mathbf{p}\in\mathcal{X}} \sqrt{\frac{1}{\sum_{t=t_0}^{T}|\mathcal{N}_t|}\sum_{t=t_0}^{T}\sum_{s\in\mathcal{N}_t}(\mathbf{p}\cdot\mathbf{x}_t^s - y_t^s)^2} \; ;$$

or simply (but many forecasters will beat it), the performance of the best model

$$\min_{j=1,\dots,N} \text{RMSE}_{t_0}((j,\dots,j)) = \min_{j=1,\dots,N} \sqrt{\frac{1}{\sum_{t=t_0}^{T}|\mathcal{N}_t|}\sum_{t=t_0}^{T}\sum_{s\in\mathcal{N}_t}\left(x_{j,t}^s - y_t^s\right)^2} \; .$$

Some notation useful for the description of the algorithms (will not be used immediately): for a vector $\mathbf{u}\in\mathbb{R}^N$ or a probability $\mathbf{p}\in\mathcal{X}$ and a round t,

$$\ell_t(\mathbf{u}) = \sum_{s\in\mathcal{N}_t}(\mathbf{u}\cdot\mathbf{x}_t^s - y_t^s)^2 \qquad \text{and} \qquad \ell_t(\mathbf{p}) = \sum_{s\in\mathcal{N}_t}(\mathbf{p}\cdot\mathbf{x}_t^s - y_t^s)^2 \; ;$$

when $\mathbf{p}$ is a Dirac mass on an expert $m = 1,\dots,N$, we simply write

$$\ell_{m,t} = \sum_{s\in\mathcal{N}_t}(x_{m,t}^s - y_t^s)^2 \; .$$

All these quantities have cumulative counterparts: for all integer $T \geq 1$,

$$L_T(\mathbf{u}) = \sum_{t=1}^{T}\ell_t(\mathbf{u}) \; , \qquad L_T(\mathbf{p}) = \sum_{t=1}^{T}\ell_t(\mathbf{p}) \; , \qquad L_{m,T} = \sum_{t=1}^{T}\ell_{m,t} \; .$$

Minimization of RMSE via the minimization of regret

We attempt to minimize the RMSE of the master forecaster by ensuring that it has a small regret, where the regret is defined either as (unconstrained prediction or convex constraints given by a convex set F)

$$\sup_{\mathbf{u}\in\mathbb{R}^N} R_T(\mathbf{u}) \text{ or } \sup_{\mathbf{u}\in F} R_T(\mathbf{u}) \qquad \text{where} \qquad R_T(\mathbf{u}) = \sum_{t=1}^{T}\sum_{s\in\mathcal{N}_t}(\mathbf{u}_t\cdot\mathbf{x}_t^s - y_t^s)^2 - \sum_{t=1}^{T}\sum_{s\in\mathcal{N}_t}(\mathbf{u}\cdot\mathbf{x}_t^s - y_t^s)^2$$

or (master predictions as convex combinations of the base predictions)

$$\max_{\mathbf{p}\in\mathcal{X}} R_T(\mathbf{p}) \qquad \text{where} \qquad R_T(\mathbf{p}) = \sum_{t=1}^{T}\sum_{s\in\mathcal{N}_t}(\mathbf{p}_t\cdot\mathbf{x}_t^s - y_t^s)^2 - \sum_{t=1}^{T}\sum_{s\in\mathcal{N}_t}(\mathbf{p}\cdot\mathbf{x}_t^s - y_t^s)^2$$

or even (still convex combinations, but we compare to single base forecasters only)

$$\max_{j=1,\dots,N} R_T(j) \qquad \text{where} \qquad R_T(j) = \sum_{t=1}^{T}\sum_{s\in\mathcal{N}_t}(\mathbf{p}_t\cdot\mathbf{x}_t^s - y_t^s)^2 - \sum_{t=1}^{T}\sum_{s\in\mathcal{N}_t}\left(x_{j,t}^s - y_t^s\right)^2 \; .$$

We can either

– minimize the regrets directly; this is, for instance, what ridge regression does, when maximizing $R_T(\mathbf{u})$ over all $\mathbf{u}\in\mathbb{R}^N$; many other forecasters, like the exponentially weighted average one, can do that on the $R_T(j)$;

– or compare to convex combinations, and first "linearize" the problems as follows, by using the convexity of $a \mapsto a^2$ (and the slope inequality),

$$R_T(\mathbf{u}) \leq \sum_{t=1}^{T} \sum_{s \in \mathcal{N}_t} 2\left(\mathbf{u}_t \cdot \mathbf{x}_t^s - y_t^s\right) \mathbf{x}_t^s \cdot (\mathbf{u}_t - \mathbf{u}) = \sum_{t=1}^{T} \mathbf{u}_t \cdot \widetilde{\ell}_t - \sum_{t=1}^{T} \mathbf{u} \cdot \widetilde{\ell}_t$$

and similarly,

$$R_T(\mathbf{p}) \leq \sum_{t=1}^{T} \mathbf{p}_t \cdot \widetilde{\ell}_t - \sum_{t=1}^{T} \mathbf{p} \cdot \widetilde{\ell}_t \leq \max_{j=1,\ldots,N} \sum_{t=1}^{T} \sum_{m=1,\ldots,N} p_{m,t} \widetilde{\ell}_{m,t} - \sum_{t=1}^{T} \widetilde{\ell}_{j,t} \ , \qquad (1)$$

where

$$\widetilde{\ell}_t = \left(\widetilde{\ell}_{1,t}, \ldots, \widetilde{\ell}_{N,t}\right) = \sum_{s \in \mathcal{N}_t} 2\left(\mathbf{v}_t \cdot \mathbf{x}_t^s - y_t^s\right) \mathbf{x}_t^s \quad (\mathbf{v}_t = \mathbf{u}_t \text{ or } \mathbf{p}_t \text{ depending on the case}). \qquad (2)$$

Numerical values

The (typical) values of the parameter on the network are

T	N	B	S	T_0	$\lvert\mathcal{N}_t\rvert$ (typ.)	L
125	48 (+3)	100	240	30	150	10^6

(where $L = \max_t \left\|\widetilde{\ell}_t\right\|_\infty$; note that $L \leq 2SB^2$ in case only predictions from the simplex are issued).

Some standard RMSE are summarized below. Mean is the master forecaster that uses $\mathbf{u}_t = (1/N, \ldots, 1/N)$ at all time rounds. FTL ("follow the leader") is the one that uses at round t the least-square estimate from time rounds 1 to $t-1$, that is,

$$\mathbf{u}_t \in \operatorname*{argmin}_{\mathbf{u} \in \mathbb{R}^N} \sum_{t'=1}^{t-1} \sum_{s \in \mathcal{N}_{t'}} \left(\mathbf{u} \cdot \mathbf{x}_{t'}^s - y_{t'}^s\right)^2 \ .$$

ELS–30 performs the same type of least-square estimation, but only over the last 30 rounds.

Best model	Mean	B_{oracle}	$B_{\text{regret}\mathbb{R}^N}$	$B_{\text{regret}\mathcal{X}}$	FTL	ELS–30
22.43	24.41	11.99	19.24	21.45	20.79	20.18

The base forecasters and the additional ELS–xx models

Here are also some statistics on the RMSE of the sample. The mean of the RMSE is

ELS–10	ELS–20	ELS–30
20.78	20.27	20.18

26.34, the median is 24.75, the maximal RMSE is 35.79 and the minimal one, 22.43. The standard deviation equals 3.69. We also give some values of the corresponding cumulative

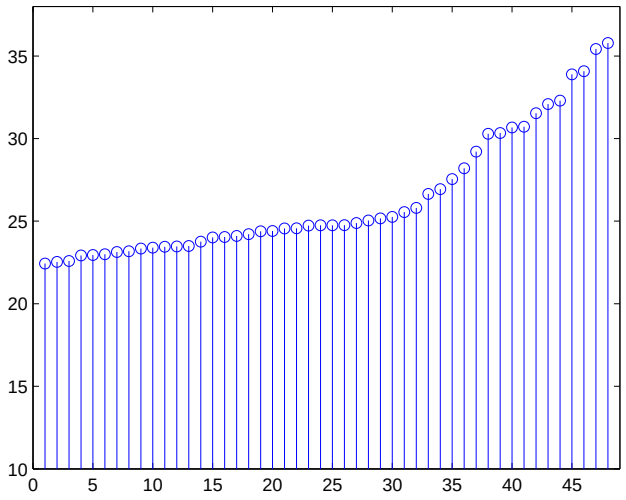

Figure 1: RMSE of the 48 base forecasters

distribution function. Denote by $F(r)$ the number of forecasters whose RMSE is less or equal to r. We have $F(23) = 6$, $F(24) = 14$, $F(25) = 27$, and $F(26) = 32$. (Out of 48 base forecasters, 32 have an RMSE less or equal to 26.) The RMSE of the additional ELS–XX forecasters are summarized above. Here is the complete (unsorted) list of the RMSE of the base forecasters.

24.01	24.40	25.80	22.59
25.04	23.50	32.30	25.55
24.57	23.00	25.17	23.18
24.10	24.89	35.79	30.29
24.75	24.04	24.75	23.76
27.55	30.72	30.67	28.20
23.40	24.73	24.21	23.45
25.26	35.42	33.89	32.09
24.39	23.34	22.43	23.47
22.53	29.21	26.65	26.94
23.13	24.56	22.92	22.95
24.76	34.08	31.54	30.34

Table 1: RMSE of the 48 base forecasters

1 Thresholded Exponentially Weighted Average Forecaster

1.1 References

Some inspiration is taken from [Allenberg et al., 2006]

1.2 Theoretical bound

[See the proof below!]
For all $\eta > 0$ and $\gamma \leq 1/(2N)$,

$$\max_{1 \leq m \leq N} R_T(\delta_m) \leq \frac{\ln N}{\eta} + \frac{\eta T}{8} S^2 B^4 + 2NSB^2 \gamma T = O(\sqrt{T})$$

where the last inequality holds for the choices of $\eta \sim 1/\sqrt{T}$ and $\gamma \leq 1/\sqrt{T}$.

1.3 Interpretation and/or comments

The forecaster below uses the standard version of the exponentially weighted average forecaster as a base routine and applies at a top level the following transformation on the prescribed convex combination: the weights less than a given threshold γ are set to zero and the remaining weights are then renormalized.

N.B.: γ should be taken less than or equal to $1/N$ to ensure that at least one weight is not less than the threshold and thus remains positive.

The algorithm is inspired by but slightly different from the one stated in [Allenberg et al., 2006] and we prove here a bound with an approximation argument. Note that the theoretical optimal parameter is $\gamma = 0$ in the current form of the bound (that is, no threshold). Still, the practical performance discussed below shows that such a threshold is valuable.

1.4 Default values of the parameters

Implemented with default values of the parameters $\eta =$ 2.2e-6 and $\gamma = 1/48$.

1.5 Statement and implementation

For $t \geq 1$, $\mathbf{p}_t$ is defined as

$$p_{m,t} = \begin{cases} c_t \, p^0_{m,t} & \text{if } p^0_{m,t} \geq \gamma \\ 0 & \text{if } p^0_{m,t} < \gamma \end{cases}$$

where

$$c_t = \frac{1}{\sum_{m=1}^N p^0_{m,t} \mathbb{I}_{\{p^0_{m,t} \geq \gamma\}}}$$

and $\mathbf{p}^0_t$ is the weight vector associated to the standard exponentially weighted average forecaster, that is $\mathbf{p}^0_1 = (1/N, \ldots, 1/N)$ and for $t \geq 2$,

$$p^0_{m,t} = \frac{\exp\left(-\eta \sum_{t'=1}^{t-1} \ell_{m,t'}\right)}{\sum_{j=1}^N \exp\left(-\eta \sum_{t'=1}^{t-1} \ell_{j,t'}\right)}$$

for all $m = 1, \ldots, N$.

Can be implemented as follows.

Parameters: learning rate η and threshold γ
Initialization: $\mathbf{p}_1^0 = (1/N, \ldots, 1/N)$
For each round $t = 1, 2, \ldots, T$,

(1) threshold the weights vector $\mathbf{p}_t^0$ into $\mathbf{p}_t$ defined as

$$p_{m,t} = \begin{cases} c_t \, p_{m,t}^0 & \text{if } p_{m,t}^0 \geq \gamma \\ 0 & \text{if } p_{m,t}^0 < \gamma \end{cases}$$

(2) predict with $\mathbf{p}_t$

(3) compute $\mathbf{p}_{t+1}^0$ as

$$p_{m,t+1}^0 = \frac{p_{m,t}^0 e^{-\eta \, \ell_{m,t}}}{\sum_{j=1}^N p_{j,t}^0 e^{-\eta \, \ell_{j,t}}}$$

for all $m = 1, \ldots, N$.

1.6 Proof of the theoretical bound

For each $t \geq 1$, we have by convexity of ℓ_t and by definition of $\mathbf{p}_t$,

$$\widehat{\ell}_t \leq \sum_{m=1}^N p_{m,t} \ell_{m,t} = \sum_{m=1}^N c_t p_{m,t}^0 \mathbb{I}_{\{p_{m,t}^0 \geq \gamma\}} \ell_{m,t} \leq c_t \sum_{m=1}^N p_{m,t}^0 \ell_{m,t} \, .$$

But

$$\frac{1}{c_t} = \sum_{m=1}^N p_{m,t}^0 \mathbb{I}_{\{p_{m,t}^0 \geq \gamma\}} = 1 - \sum_{m=1}^N p_{m,t}^0 \mathbb{I}_{\{p_{m,t}^0 < \gamma\}} \geq 1 - N\gamma \, .$$

Hence, using that $N\gamma < 1/2$, we get

$$c_t \leq \frac{1}{1 - N\gamma} \leq 1 + 2N\gamma \, .$$

We thus have upper-bounded $\widehat{\ell}_t$ as

$$\widehat{\ell}_t \leq (1 + 2N\gamma) \sum_{m=1}^N p_{m,t}^0 \ell_{m,t} \, .$$

Summing over all time rounds $t = 1, \ldots, T$, substracting the cumulative loss of the best model, and substituting the bound on the exponentially weighted average forecaster (see, e.g., [Mallet et al., 2007]), we get

$$\begin{aligned} \widehat{L}_T - \min_{1 \leq j \leq T} L_{j,T} &\leq \left(\sum_{t=1}^T \sum_{m=1}^N p_{m,t}^0 \ell_{m,t} - \min_{1 \leq j \leq T} L_{j,T} \right) + 2N\gamma \sum_{t=1}^T \sum_{m=1}^N p_{m,t}^0 \ell_{m,t} \\ &\leq \frac{\ln N}{\eta} + \frac{\eta \, T}{8} S^2 B^4 + 2N\gamma T S B^2 \, . \end{aligned}$$

1.7 Performance

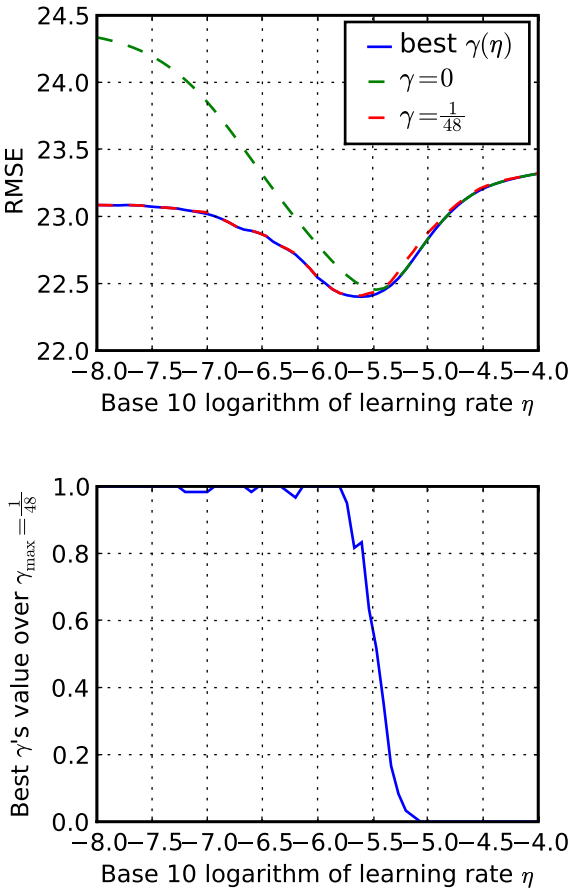

Figure 2: Empirical performance: RMSE versus η for minimum, maximum, and best threshold values (top); ratio of the best threshold $\gamma(\eta)$ for each η, over its maximum value $1/48$ (bottom).

	$\eta = 3.4\text{e-}6$ $\gamma = 0$	$\eta = 2.2\text{e-}6$ $\gamma = 1/48$	$\eta^* = 2.5\text{e-}6$ $\gamma^* = 0.83/48$
RMSE	22.46	22.41	22.40

Thresholding results in a slight improvement in the RMSE. The maximal thresholding level is almost as good as the best empirical thresholding level.

The interest of thresholding lies rather in an improved robustness against the choice of the learning parameter η. Figure 2 shows that the RMSE curve is flatter in presence of the maximal threshold. Actually, differences between the two curves appear mostly for parameters η smaller than the optimal value η^*. Learning parameters η larger than η^* give already rise to an drastic selection of the models, and thus, a further selection by thresholding is useless. On the contrary, too small an η does not lead to the selection

of a subset of models and thresholding forces such a selection in this case; empirical performance shows that this additional thresholding has a positive impact on the RMSE.

1.8 Graphical evolution of the weights

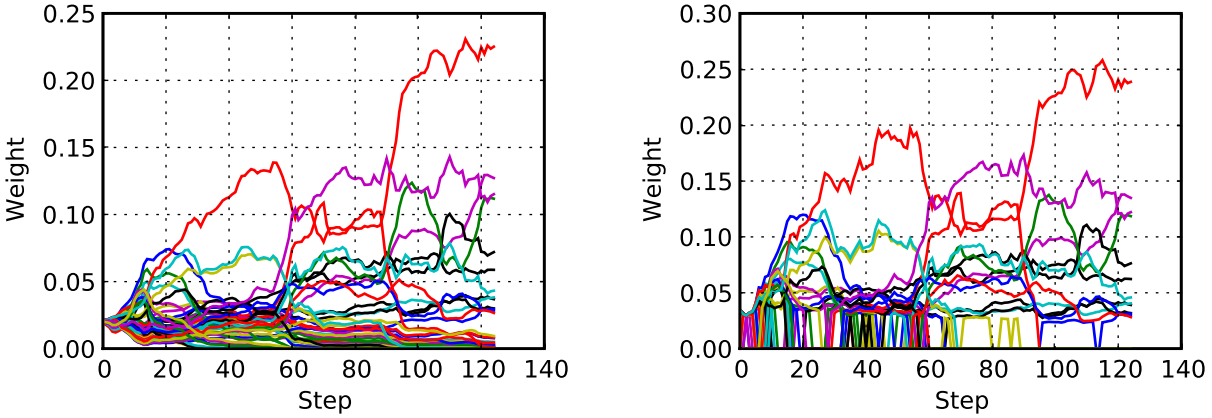

Figure 3: Evolution of the weights for $\eta = 2.2\text{e-}6$ and for $\gamma = 0$ (left) and $\gamma = 1/48$ (right)

2 Thresholded Exponentiated Gradient

2.1 References

Some inspiration is taken from [Allenberg et al., 2006]

2.2 Theoretical bound

[See the proof below!]
For all $\eta > 0$ and $\gamma \leq 1/(2N)$,

$$\max_{\mathbf{p}\in\mathcal{X}} R_T(\mathbf{p}) \leq \frac{\ln N}{\eta} + \frac{\eta T}{2} L^2 + 2NL\gamma T = O(\sqrt{T})$$

where the last inequality holds for the choices of $\eta \sim 1/\sqrt{T}$ and $\gamma \leq 1/\sqrt{T}$.

2.3 Interpretation and/or comments

The same comments as in Section 1 apply; except that here, the master forecaster is the linearized version of the exponentially weighted average forecaster, called the exponentiated gradient forecaster.

2.4 Default values of the parameters

Implemented with default values of the parameters $\eta = 2.1\text{e-}5$ and $\gamma = 1/48$.

2.5 Statement and implementation

For $t \geq 1$, $\mathbf{p}_t$ is defined as

$$p_{m,t} = \begin{cases} c_t\, p^0_{m,t} & \text{if } p^0_{m,t} \geq \gamma \\ 0 & \text{if } p^0_{m,t} < \gamma \end{cases}$$

where

$$c_t = \frac{1}{\sum_{m=1}^N p^0_{m,t} \mathbb{I}_{\{p^0_{m,t}\geq\gamma\}}}$$

and $\mathbf{p}^0_t$ is the weight vector associated to Exponentiated Gradient, that is, $\mathbf{p}^0_1 = (1/N, \ldots, 1/N)$ and for $t \geq 2$,

$$p^0_{m,t} = \frac{\exp\left(-\eta \sum_{t'=1}^{t-1} \widetilde{\ell}_{m,t'}\right)}{\sum_{j=1}^N \exp\left(-\eta \sum_{t'=1}^{t-1} \widetilde{\ell}_{j,t'}\right)}$$

for all $m = 1, \ldots, N$, where we recall that the $\widetilde{\ell}_{j,t'}$ were defined in (2).

Can be implemented as follows.

Parameters: learning rate η and threshold γ
Initialization: $\mathbf{p}_1^0 = (1/N, \ldots, 1/N)$
For each round $t = 1, 2, \ldots, T$,

(1) threshold the weights vector $\mathbf{p}_t^0$ into $\mathbf{p}_t$ defined as

$$p_{m,t} = \begin{cases} c_t \, p_{m,t}^0 & \text{if } p_{m,t}^0 \geq \gamma \\ 0 & \text{if } p_{m,t}^0 < \gamma \end{cases}$$

(2) predict with $\mathbf{p}_t$

(3) compute $\mathbf{p}_{t+1}^0$ as

$$p_{m,t+1}^0 = \frac{p_{m,t}^0 e^{-\eta \widetilde{\ell}_{m,t}}}{\sum_{j=1}^N p_{j,t}^0 e^{-\eta \widetilde{\ell}_{j,t}}}$$

for all $m = 1, \ldots, N$.

2.6 Proof of the theoretical bound

We first have by convexity (the so-called slope inequality), like in (1), that

$$\widehat{L}_T - \min_{\mathbf{p} \in \mathcal{X}} L_T(\mathbf{p}) \leq \max_{j=1,\ldots,N} \sum_{t=1}^T \sum_{m=1,\ldots,N} p_{m,t} \widetilde{\ell}_{m,t} - \sum_{t=1}^T \widetilde{\ell}_{j,t}$$

and can thus apply the same argument as in Section 1 with the only replacement of the $\ell_{m,t}$ (bounded between 0 and SB^2) there by the $\widetilde{\ell}_{m,t}$ (bounded between $-L$ and L).

2.7 Performance

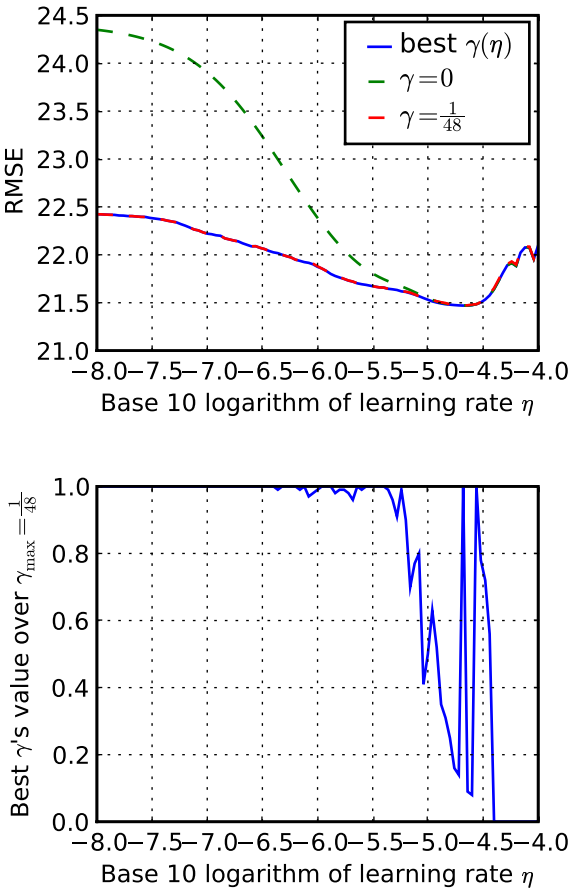

Figure 4: Practical performance with the N base forecasters: RMSE versus η for minimum, maximum and best threshold values (upper); ratio of the best γ's value w.r.t. η over its maximum value (lower)

The best RMSE equals 21.47 and is obtained for the (a posteriori) choices of $\eta^* \approx$ 2.1e-5 and $\gamma^* = 1/48$ (the maximal threshold level). This RMSE does not differ by more than 5e-3 from the RMSE of the unthresholded version. Thus, up to accuracy issues, no improvement is gained from thresholding in this case as far as the RMSE is concerned.

However, the main improvement of thresholding is a gain in robustness against a bad (too small) choice of the learning parameter. The same comments as in Section 1 apply here again.

2.8 Graphical evolution of the weights

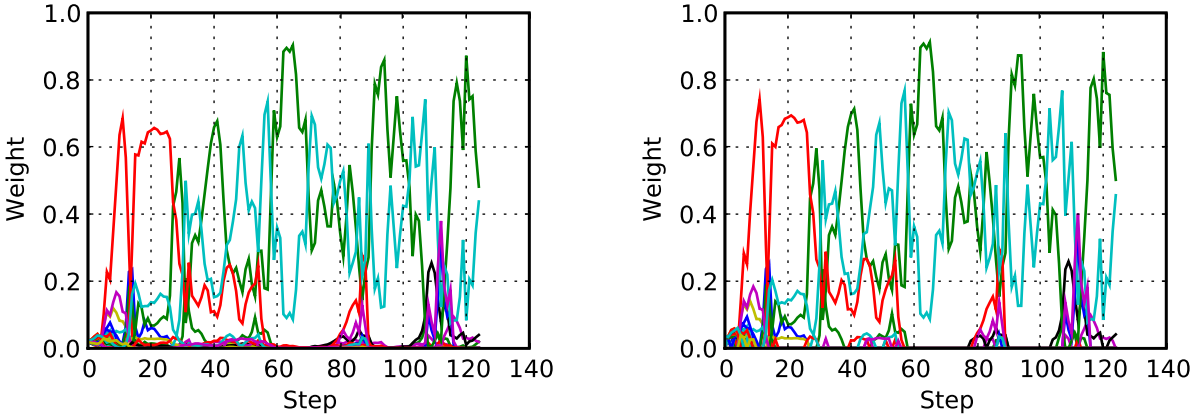

Figure 5: Evolution of the weights for η =2.09e-5 and for $\gamma = 0$ (left) or $\gamma = \frac{1}{N}$ (right)

3 Lasso

3.1 References

The Lasso method was introduced in [Tibshirani, 1996]. A thorough investigation of the Lasso problem and its dual was given in [Osborne et al., 2000b], following an article by the same authors giving a first homotopy method to compute all the Lasso solutions at once [Osborne et al., 2000a]. The LARS-type implementation of Lasso was described in [Efron et al., 2004].

3.2 Theoretical bound

No individual sequence-type regret bound for the Lasso has been proven yet. However, some convergence rates were obtained in stochastic settings. The special case of linear regression on a fixed design is, for example, derived from oracle inequalities for the prediction loss in [Bickel et al., 2008].

Let $y = (y_1, \dots, y_n) = X\beta^* + W$ be a vector of n realizations of the response variable y, where X is a deterministic $n \times M$ matrix, $\beta^* \in \mathbb{R}^M$ and $W = (W_1, ..., W_n)$ a vector of i.i.d. $\mathcal{N}(0, \sigma^2)$ variables. The Lasso estimator is defined by

$$\widehat{\beta} = \underset{\beta \in \mathbb{R}^M}{\operatorname{argmin}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(y_i - (X\beta)_i\right)^2 + 2r \sum_{j=1}^{M} |\beta_j| \right\} .$$

Then, if all the diagonal elements of the matrix $X^T X/n$ equal 1 and under certain conditions described in [Bickel et al., 2008], for r of the order of $\sqrt{(\log M)/n}$, one has with large probability that

$$\frac{1}{n} \sum_{i=1}^{n} \left(X(\widehat{\beta} - \beta^*)\right)_i^2 = \mathcal{O}\left(\frac{\log M}{n}\right) .$$

Other results dealing with prediction loss and/or excess risks may be found in [Bunea et al., 2007] and [de Geer, 2008]. The proven upper bounds are again of the order of $(\log M)/n$.

3.3 Statement, interpretation and/or comments

The version of the Lasso parameterized by $\lambda \geq 0$ chooses $\mathbf{u}_1 = (0, \dots, 0)$ and for $t \geq 2$,

$$\mathbf{u}_t(\lambda) \in \underset{\mathbf{u} \in \mathbb{R}^N}{\operatorname{argmin}} \left[\lambda \|\mathbf{u}\|_1 + \sum_{t'=1}^{t-1} \ell_{t'}(\mathbf{u}) \right] = \underset{\mathbf{u} \in \mathbb{R}^N}{\operatorname{argmin}} \left[\lambda \|\mathbf{u}\|_1 + \sum_{t'=1}^{t-1} \sum_{s \in \mathcal{N}_{t'}} (\mathbf{u} \cdot \mathbf{x}_{t'}^s - y_{t'}^s)^2 \right]$$

The Lasso thus appears as a ℓ^1-penalized "follow the leader" (sequential least-squares regression) forecaster. It enjoys both the shrinkage property of the ridge regression forecaster (ℓ^2-penalty) and variable selection, thus leading to improved prediction accuracy and better interpretability.

The LARS-Lasso algorithm proposed in [Efron et al., 2004] enables to efficiently compute all the Lasso solutions, i.e., the whole path $\lambda \mapsto \mathbf{u}_t(\lambda)$, at each round $t \geq 2$. Since these paths can be shown to be continuous and piecewise-linear (and constant for λ large

enough), they are completely described by the breakpoint values, denoted by $\lambda_{t,k}$, and the corresponding coefficient vectors $\mathbf{u}_{t,k} \in \mathbb{R}^N$.

Note that this algorithm assumes the *one-at-a-time condition* — see [Efron et al., 2004]. Since its termination property has not been proven yet, one could add a stopping criterion.

We also highlight a numerical issue: the so-called *numerical precision error* has a major influence on the (theoretically valid) algorithm sketched below. This is mainly due to comparison tests (among which, sign tests), whose results can be incorrect in practice. We thus recommand to adapt the implementation by considering the three following points:

- explicitly put the components of the coefficient vector $\mathbf{u}_{t,k}$ to zero when theoretically known to equal zero,
- omit terms known to (theoretically) equal zero within the formula defining $\widehat{\gamma}_k$,
- compute the maximal absolute correlation $\widehat{C}_k$ directly instead of using the seemingly more efficient recursive formulation $\widehat{C}_{k+1} = \widehat{C}_k - \widehat{\gamma}_{k+1} A_{k+1}$.

3.4 Default value of the parameter

Implemented with default value of the parameter $\lambda = 300$.

3.5 Implementation

As mentioned in Section 3.3, it suffices to compute, at each round $t \geq 2$, the breakpoint values $\lambda_{t,k}$ and the corresponding coefficient vectors $\mathbf{u}_{t,k} \in \mathbb{R}^N$ of the continuous and piecewise-linear path $\lambda \mapsto \mathbf{u}_t(\lambda)$, for $0 \leq k \leq K_t$. (We sort the $\lambda_{t,k}$ in a non-increasing fashion.) This is what the LARS-Lasso algorithm implements. Given those values and the chosen parameter value λ, it then simply remains to

- compute the linear interpolation $\mathbf{u}_t$ between $(\lambda_{t,k+1}, \mathbf{u}_{t,k+1})$ and $(\lambda_{t,k}, \mathbf{u}_{t,k})$, where $\lambda \in [\lambda_{t,k+1}, \lambda_{t,k}[$, if $\lambda < \lambda_{t,0} = \max_k \lambda_{t,k}$,
- let $\mathbf{u}_t = \mathbf{u}_{t,0} = (0, \ldots, 0)$, if $\lambda \geq \lambda_{t,0}$.

To do so, at each round $t \geq 2$, the LARS-Lasso forecaster proceeds in loops and uses the available data only through the following two quantities (both summations only go till $t-1$):

$$G_t = \sum_{t'=1}^{t-1} \sum_{s \in \mathcal{N}_{t'}} \mathbf{x}_{t'}^s \left(\mathbf{x}_{t'}^s\right)^\top \quad \text{and} \quad \mathbf{c}_t^0 = \sum_{t'=1}^{t-1} \sum_{s \in \mathcal{N}_{t'}} y_{t'}^s \mathbf{x}_{t'}^s \ .$$

G_t is the Gram matrix of the covariates given the data till round $t-1$, while $\mathbf{c}_t^0$ is the empirical correlation vector (up to a multiplicative factor) between the covariates and the observation vector on the same time range.

At the end of the k-th LARS-Lasso step, the set $\mathcal{A}_{t,k} \subset \{1, \ldots, N\}$ will denote the new active set used at the next step, $c_{t,k}$, the current correlation vector (up to a multiplicative factor) between the covariates and the still unexplained part of the observations

$$\mathbf{c}_{t,k} = \mathbf{c}_t^0 - G_t \mathbf{u}_{t,k} \ ,$$

$\mathbf{s}_{t,k} = (\operatorname{sgn} \mathbf{c}_{t,k,m})_{1\leq m\leq N} \in \{-1,0,1\}^N$, the signs of its components, and $\lambda_{t,k} = \|\mathbf{c}_{t,k}\|_\infty$, the maximal absolute correlation (corresponding to the quantity referred to as $\hat{C}_{k+1}$ in [Efron et al., 2004]).

The whole algorithm can be implemented as is described below by using the following additional notation. For a given matrix X and two subsets of indexes $\mathcal{A}_1, \mathcal{A}_2 \subset \{1,\dots,N\}$, we denote by $X_{|\mathcal{A}_1,\mathcal{A}_2}$ the matrix extracted of X by picking only rows with indexes in $\mathcal{A}_1$ and columns with indexes in $\mathcal{A}_2$; the '$\cdot$' symbol refers to the whole set $\{1,\dots,N\}$ and thus enables to perform only row-wise or column-wise extractions.

Parameters: regularization parameter $\lambda \geq 0$

Initialization: $\mathbf{u}_1 = (0,\dots,0)$, $G_1 = 0$, $\mathbf{c}_1^0 = (0,\dots,0)$

For each round $t = 1,2,\dots,n$,

(1) predict with $\mathbf{u}_t$;

(2) get the observations and compute

$$G_{t+1} = G_t + \sum_{s\in\mathcal{N}_t} \mathbf{x}_t^s (\mathbf{x}_t^s)^\top \quad \text{and} \quad \mathbf{c}_{t+1}^0 = \mathbf{c}_t^0 + \sum_{s\in\mathcal{N}_t} y_t^s \mathbf{x}_t^s \ ;$$

(3) go through the LARS-Lasso loops:

Loop initialization:
$k = 0$, $\mathbf{c}_{t+1,0} = \mathbf{c}_{t+1}^0$, $\mathbf{s}_{t+1,0} = \operatorname{sgn}(\mathbf{c}_{t+1}^0)$, $\lambda_{t+1,0} = \left\|\mathbf{c}_{t+1}^0\right\|_\infty$,
$\mathcal{A}_{t+1,0} = \operatorname{argmax}_{1\leq m\leq N} \left|c_{t+1,m}^0\right|$;
raise an error if $\#\mathcal{A}_{t+1,0} > 1$

While $(\#\mathcal{A}_{t+1,k} < N)$ or $(\#\mathcal{A}_{t+1,k} = N$ and $\#\mathcal{A}_{t+1,k-1} < N)$

(a) compute

* $\mathcal{G}_{t+1,k+1} = G_{t+1|\mathcal{A}_{t+1,k},\mathcal{A}_{t+1,k}}$,
* $\tilde{w}_{t+1,k+1} = \mathcal{G}_{t+1,k+1}^{-1} \mathbf{s}_{t+1,k}$,
* $A_{t+1,k+1} = \left(\mathbf{s}_{t+1,k}^\top \tilde{w}_{t+1,k+1}\right)^{-1/2}$ (possible, by positiveness and symmetry),
* $w_{t+1,k+1} = A_{t+1,k+1} \tilde{w}_{t+1,k+1}$,
* $a_{t+1,k+1} = G_{t+1|\cdot,\mathcal{A}_{t+1,k}}$;

(b) compute ($\min^+$ stands for minimum over positive arguments only)

$$\hat{\gamma}_{t+1,k+1} = \min_{m\in\mathcal{A}_{t+1,k}^c}{}^{+} \left\{ \frac{\lambda_{t+1,k} - c_{t+1,k,m}}{A_{t+1,k+1} - a_{t+1,k+1,m}} \ , \ \frac{\lambda_{t+1,k} + c_{t+1,k,m}}{A_{t+1,k+1} + a_{t+1,k+1,m}} \right\}$$

and the unique corresponding minimizer $\hat{m}$ (if more than one, raise an error);

(c) compute ($1/0 = +\infty$ by convention)

$$\widetilde{\gamma}_{t+1,k+1} = \min_{m \in \mathcal{A}_{t+1,k}}^{+} \left\{ \frac{-u_{t+1,k,m}}{w_{t+1,k+1,m}} \right\}$$

and the possible minimizer $\widetilde{m}$ (if more than one, raise error);

(d) if $\widetilde{\gamma}_{t+1,k+1} < \widehat{\gamma}_{t+1,k+1}$, let $\mathcal{A}_{t+1,k+1} = \mathcal{A}_{t+1,k} - \{\widetilde{m}\}$ and $\widehat{\gamma} = \widetilde{\gamma}$; else, let $\mathcal{A}_{t+1,k+1} = \mathcal{A}_{t+1,k} \cup \{\widehat{m}\}$;

(e) compute $u_{t+1,k+1,m} = u_{t+1,k+1,m} + \widehat{\gamma}_{t+1,k+1} w_{t+1,k+1,m}$ for $m \in \mathcal{A}_{t+1,k+1}$ and $u_{t+1,k+1,m} = 0$ otherwise;

(f) compute $c_{t+1,k+1} = c_{t+1,k} - \widehat{\gamma}_{t+1,k+1} a_{t+1,k+1}$;

(g) update $s_{t+1,k}$ to $s_{t+1,k+1}$ by recomputing the sign for the added index (if any);

(h) compute $\lambda_{t+1,k+1} = \|\mathbf{c}_{t+1,k+1}\|_\infty$;

(i) set $k = k + 1$;

(4) compute the linear interpolation given λ,

$$\mathbf{u}_{t+1} = \begin{cases} \dfrac{\lambda - \lambda_{t+1,k+1}}{\lambda_{t+1,k} - \lambda_{t+1,k+1}} \left(\mathbf{u}_{t+1,k} - \mathbf{u}_{t+1,k+1} \right) & \text{if } \lambda_{t+1,k+1} \leq \lambda < \lambda_{t+1,k}, \\ (0, \ldots, 0) & \text{if } \lambda \geq \lambda_{t+1,0}. \end{cases}$$

3.6 Performances

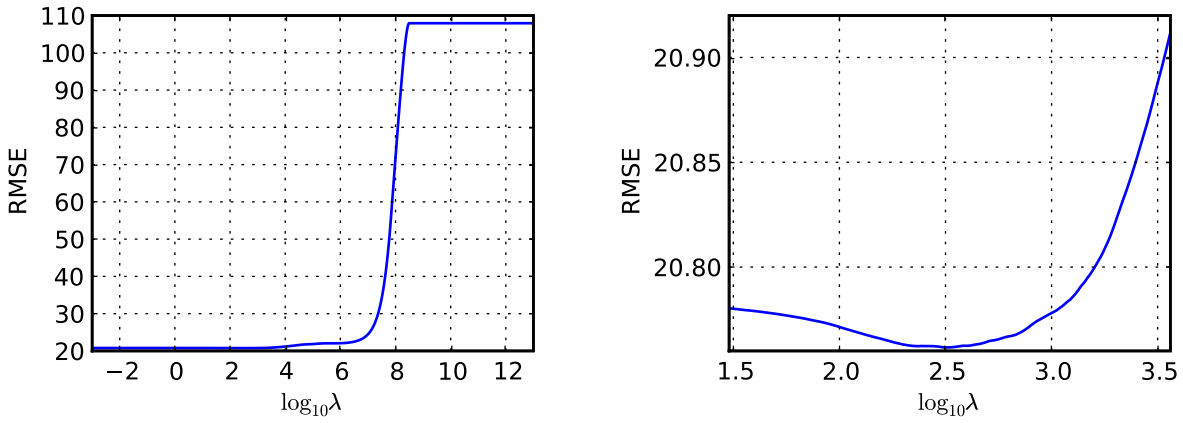

Figure 6: Practical performances with the N base forecasters: RMSE versus λ for the whole test range (left) and around the minimum (right).

The best RMSE equals 20.76 and is obtained for the (a posteriori) choice of $\lambda^* \approx 320$. The Lasso thus has about the same best practical performance as the ridge regression forecaster (20.77 on the same data). Note that this best RMSE is already almost achieved

by the "follow the leader" forecaster (20.79) as well.

As far as robustness is concerned, the optimal regularization parameter λ^* is nearly at the utmost right of the left flat region of the RMSE curve. This suggests slightly under-penalizing to avoid rapidly increasing RMSE regions. The same remark holds for variants of the Lasso — see, e.g., Section 4.

We also note that the Lasso has constant RMSE for λ large enough. For theses values, the constant weight $\mathbf{u}_t = (0, \ldots, 0)$ indeed solves the minimization problem, which is not the case for the ridge regression forescater which only ensures that $\mathbf{u}_t \to (0, \ldots, 0)$ as $\lambda \to \infty$. Future attempts to calibrate λ can thus be restricted on a bounded interval, which is very convenient compared to, e.g., ridge regression.

3.7 Variable selection

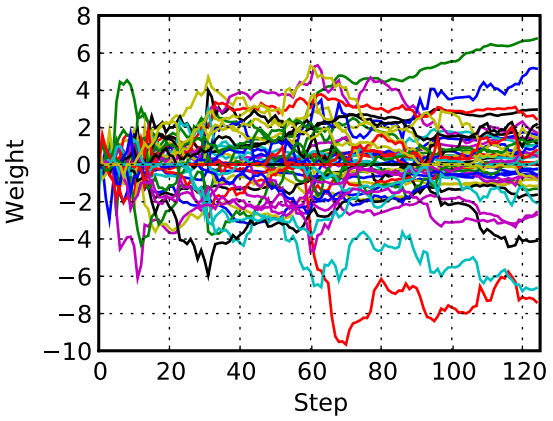

Figure 7: Evolution of the weights for $\lambda = \lambda^* \approx 320$.

Figure 7 shows that some weights strictly equal zero at some time instants. The Lasso thus acts as a variable selector, which is an important characteristic beyond coefficient shrinkage.

More precisely, the number of zero weights globally decreases with time as shown on Figure 8. This is quite natural since the penalty term has less influence as the number of terms in the cumulative loss term increases (hence leading to a loosened constraint).

Figure 8 also shows that all the base forecasters are selected, at least at two or three time instants, by the Lasso (i.e., get non-zero weight). This tends to prove that the whole ensemble is useful for the Lasso.

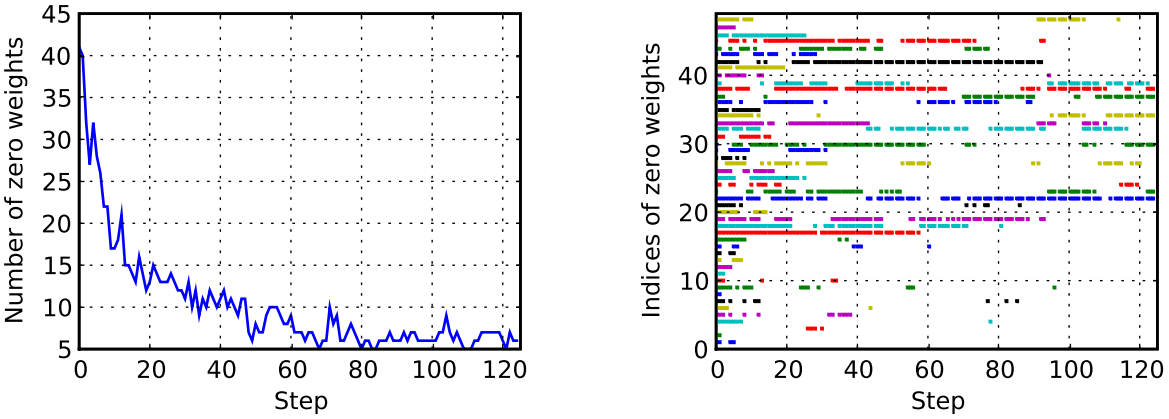

Figure 8: Evolution of the number of zero weights (left) and corresponding indexes (right) for $\lambda = \lambda^* \approx 320$.

Finally, Figure 9 illustrates the way the Lasso selects the base forecasters. The low-RMSE ones are quite always kept, but the large-RMSE ones can be left out (get zero weight) or kept. The Lasso is right to keep them if, for instance, they only differ from the true observations by a constant multiplicative factor, from which the weight put on the model should then be close to. (This is where the fact that the combination is linear and not necessarily convex plays a crucial role.)

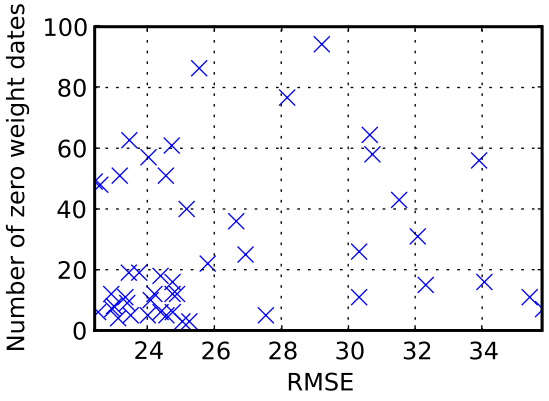

Figure 9: Role of RMSE in the Lasso selection for $\lambda = 300$. Each of the 48 base forecasters is plotted by two coordinates: its RMSE_{t_0} (x-axis), and the number of time instants the Lasso puts its weight to zero (y-axis).

4 Lasso with Discounted Losses

4.1 References

This is a new forecaster. It is strongly motivated by the good performance shown for the discounted version of the ridge regresssion forecaster.

4.2 Theoretical bound

None yet! But it should be easy to obtain one by approximation arguments once an individual sequence-type regret bound for the Lasso has been proven.

4.3 Interpretation and/or comments

A version of the Lasso corrected by discount factors that puts more weight on more recent losses.

4.4 Default values of the parameters

Implemented with default values of the parameters $\lambda = 2.e4$, $\beta = 1.5$, and $\gamma = 150$.

4.5 Statement and implementation

Consider a decreasing sequence of positive numbers $(\beta_t)_{t\geq 1}$. We take it depending on two parameters $\gamma > 0$ and $\beta > 0$ only, say $\beta_t = 1 + \gamma/t^\beta$, just as for the discounted ridge regression forecaster. The version of the discounted Lasso parameterized by $\lambda \geq 0$ chooses $\mathbf{u}_1 = (0, \ldots, 0)$ and, for $t \geq 2$,

$$\begin{aligned} \mathbf{u}_t(\lambda;\beta,\gamma) &\in \operatorname*{argmin}_{\mathbf{u}\in\mathbb{R}^N} \left[\lambda \|\mathbf{u}\|_1 + \sum_{t'=1}^{t-1} \beta_{t-t'}\, \ell_{t'}(\mathbf{u}) \right] \\ &= \operatorname*{argmin}_{\mathbf{u}\in\mathbb{R}^N} \left[\lambda \|\mathbf{u}\|_1 + \sum_{t'=1}^{t-1} \beta_{t-t'} \sum_{s\in\mathcal{S}_{t'}} \left(\mathbf{u}\cdot\mathbf{x}_{t'}^s - y_{t'}^s\right)^2 \right] . \end{aligned}$$

Just as for the standard Lasso, the LARS-Lasso algorithm enables to efficiently compute the whole discounted Lasso paths. At each round $t \geq 2$, the algorithm proceeds exactly in the same way, after transforming the observations and the covariates into their discounted counterparts. More precisely, the two quantities G_t and $\mathbf{c}_t^0$ are now defined by

$$G_t = \sum_{t'=1}^{t-1} \beta_{t-t'} \sum_{s\in\mathcal{S}_{t'}} \mathbf{x}_{t'}^s \left(\mathbf{x}_{t'}^s\right)^\top \quad \text{and} \quad \mathbf{c}_t^0 = \sum_{t'=1}^{t-1} \beta_{t-t'} \sum_{s\in\mathcal{S}_{t'}} y_{t'}^s \mathbf{x}_{t'}^s$$

(note that both summations only go till $t-1$). This time, there is no efficient sequential update for these quantities.

The implementation of the discounted Lasso is thus point-by-point similar to the standard Lasso implementation, except that there are now three parameters ($\lambda \geq 0$, $\gamma > 0$ and $\beta > 0$) and that step (2) must be replaced by

(2) get the observations and compute

$$G_{t+1} = \sum_{1 \le t' \le t} \beta_{t+1-t'} \sum_{s \in \mathcal{N}_{t'}} x^s_{t'} \left(x^s_{t'}\right)^\top \quad \text{and} \quad c^0_{t+1} = \sum_{1 \le t' \le t} \beta_{t+1-t'} \sum_{s \in \mathcal{N}_{t'}} y^s_{t'} x^s_{t'} \, ;$$

4.6 Performance

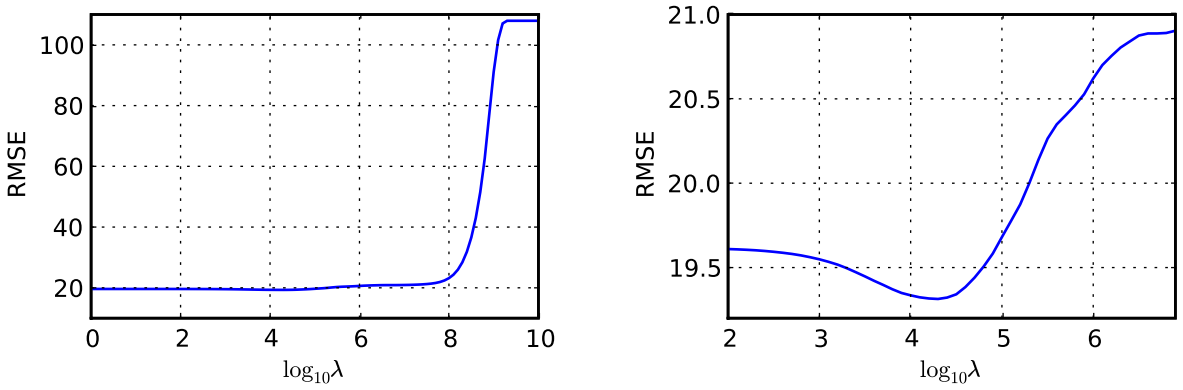

Figure 10: Practical performance with the N base forecasters and the discount factors $\beta = \beta^* \approx 1.5$ and $\gamma = \gamma^* \approx 150$: RMSE versus λ for the whole test range (left) and around the minimum (right).

Simulations on a (λ, β, γ)-mesh suggest that the best RMSE approximately equals 19.31 and is obtained for the (a posteriori) choice of $\lambda^* \approx$ 2.e4, $\beta^* \approx 1.5$, $\gamma^* \approx 150$. The discounted Lasso thus has about the same best practical performance as the discounted ridge regression forecaster (19.38 on the same data and with the same discount factors).

Recall that, with non-discounted losses, the "follow the leader" forecaster (sequential least squares) performed almost as well as the ridge regression forecaster and the Lasso. The situation is here a bit different with discounted losses, and the ℓ^1-penalty shows some effectiveness (0.3 absolute decrease in the RMSE w.r.t. the discounted "follow the leader" with the same discount factors). The same remark held for the ℓ^2-penalty, but the discounted Lasso also proves to be quite a strong variable selector — see Section 4.7.

We also note the influence of the discount factors on the RMSE. Figure 11 shows that the larger the discount amplitude γ (or the lower the discount exponent β), the larger the optimal regularization parameter $\lambda^*(\beta, \gamma) \in \operatorname{argmin}_{\lambda \ge 0} \text{RMSE}_{t_0}(\beta, \gamma)$. This is quite natural since the cumulative loss term in the criterion minimized by the Lasso is then larger.

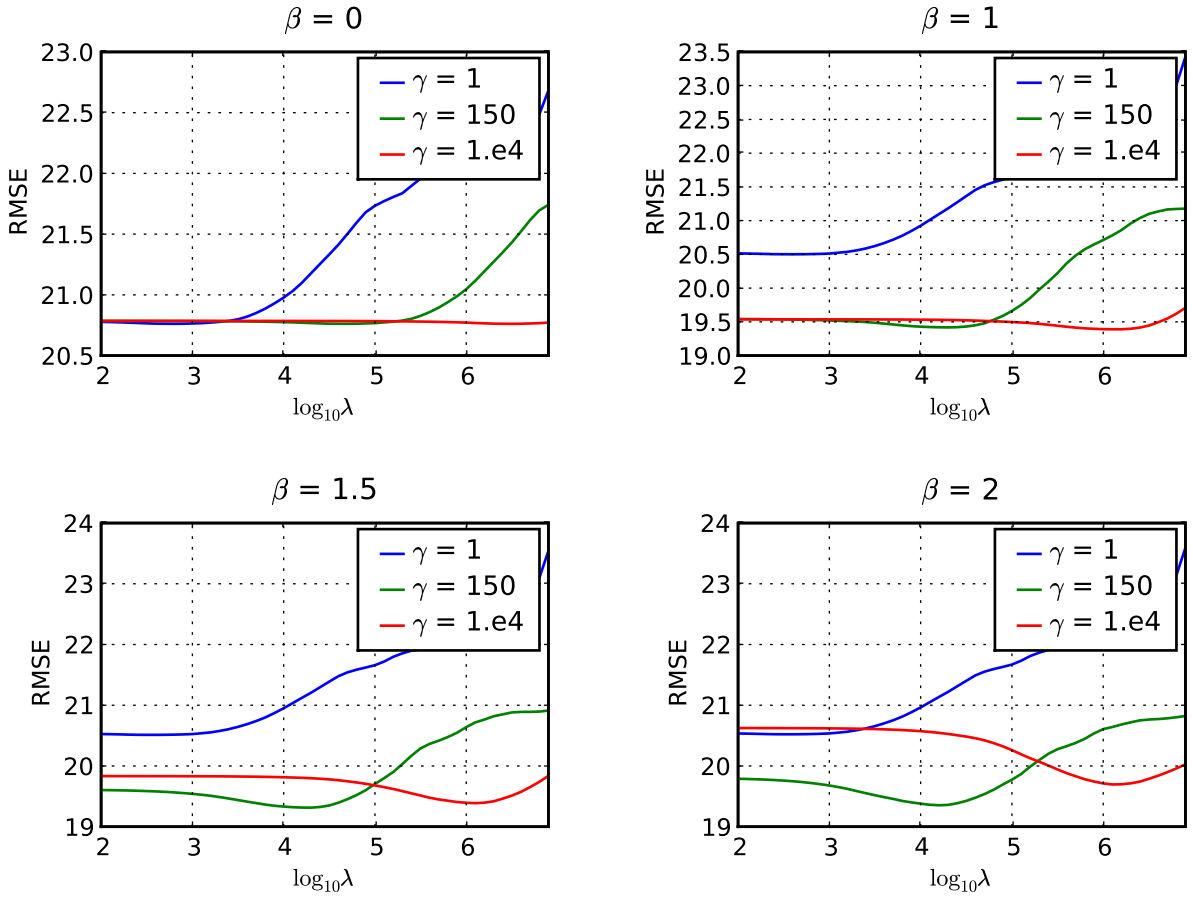

Figure 11: Practical performance with the N base forecasters for different values of the regularization parameter λ and the discount factors β and γ. The parameter β is constant on each single graph and respectively equals 0 (upper left), 1 (upper right), 1.5 (lower left) and 2 (lower right).

Since the discounted Lasso is the standard Lasso applied to the "discounted" data, two further comments on the standard Lasso hold as well. To ensure good performance with sufficient robustness, we thus suggest slightly under-penalizing to avoid the part of the RMSE curve where it rapidly increases.

Idem for the calibration remark — the optimal regularization parameter λ can be studied in a bounded interval.

4.7 Variable selection

Figures 12 and 13 shows the weights evolution for the optimal parameters. The average number of zero weights on the "stationary" period (60 first days excluded) approximately equals 17.6 . The discounted Lasso thus proves to be an even better variable selector than the base version (6.5 zero weights on the average for $\lambda = 320$). This is quite natural since

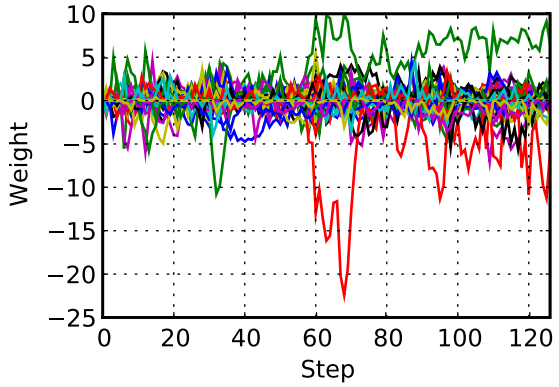

Figure 12: Evolution of the weights for $\lambda = 2.e4$, $\beta = 1.5$, and $\gamma = 150$.

it is more easy to discriminate between the base forecasters on a small period (the recent past).

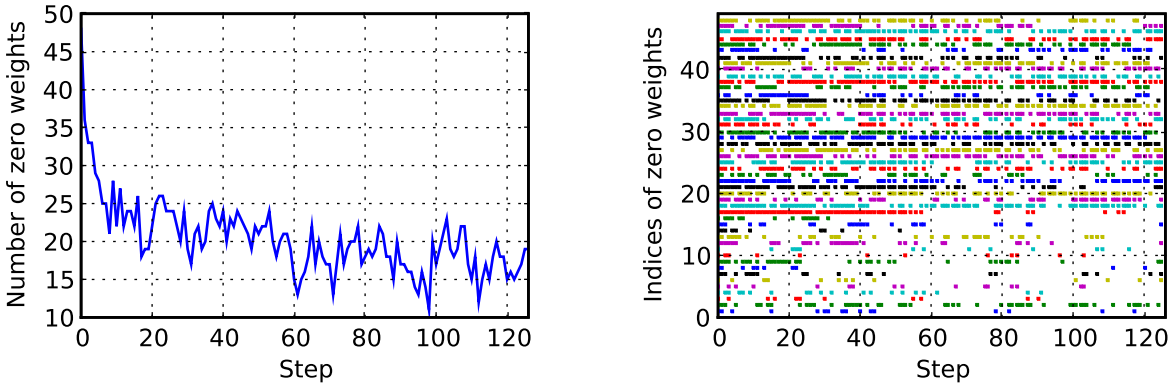

Figure 13: Evolution of the number of zero weights (left) and corresponding indexes (right) for $\lambda = 2.e4$, $\beta = 1.5$, and $\gamma = 150$.

The same remarks on the globally decreasing property of the number of zero weights, or on the richness of the base forecasters ensemble, hold as in the no-discount case.

We eventually note that the larger the discount amplitude γ (or the lower the exponent β), the weaker the variable selection — see Figure 14. This is again quite natural, since, with constant λ, the cumulative loss term is then larger and leads to a loosened constraint.

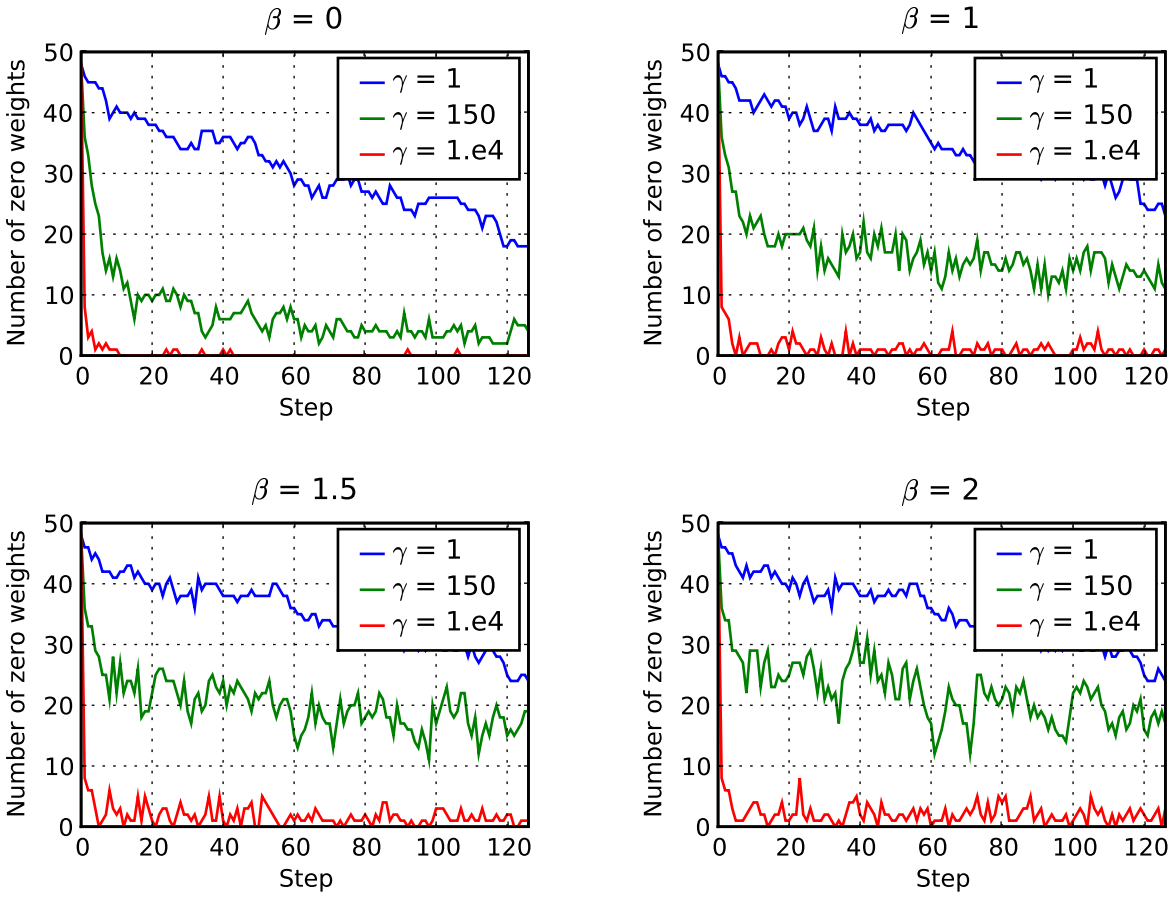

Figure 14: Evolution of the number of zero weights for $\lambda = 2.e4$ and for different values of the discount factors β and γ. The parameter β is constant on each single graph and respectively equals 0 (upper left), 1 (upper right), 1.5 (lower left) and 2 (lower right).

5 Renormalized Lasso

5.1 References

This is a new variant of the Lasso that is defined for both undiscounted or discounted losses. It is motivated by intuitive considerations — see Section 5.3 — and stochastic-type theoretical results — see Section 5.2.

5.2 Theoretical bound

None yet! But, as mentioned in 3.2 to motivate the standard Lasso, some convergence rates were proven in stochastic settings. Under assumptions of the kind of those described therein, with large probability, the prediction loss (or the conditional excess risk) is of the order of $(\log M)/n$, if the regularization parameter r after renormalizing the cumulative loss term is taken of the order of $\sqrt{(\log M)/n}$.

5.3 Statement, interpretation and/or comments

The "renormalized Lasso" is defined as a generic modification of the Lasso for both undiscounted or discounted losses. It consists in refining the Lasso penalty term

$$\lambda \|u\|_1$$

into what we will call the "renormalized Lasso penalty" term,

$$\lambda \left(\sum_{t'=1}^{t-1} |\mathcal{N}_{t'}| \right)^{\alpha} \|u\|_1 \ ,$$

for the renormalized non-discounted version of the Lasso, or

$$\lambda \left(\sum_{t'=1}^{t-1} \beta_{t-t'} \, |\mathcal{N}_{t'}| \right)^{\alpha} \|u\|_1 \ ,$$

for the renormalized discounted Lasso, where the sequence $(\beta_t)_{t\geq 1} = \left(1+\gamma/t^{\beta}\right)_{t\geq 1}$ is defined as in Section 4.5.

This new penalty term thus takes into account the (possibly discounted) total number of observations seen so far, $|\mathcal{N}_1| + \ldots + |\mathcal{N}_{t-1}|$ (or $\beta_{t-1}|\mathcal{N}_1| + \ldots + \beta_1 |\mathcal{N}_{t-1}|$ in case of discounted losses) and prevents the λ penalty term from becoming negligible as t increases. We chose to consider an additional parameter α, which controls the growth rate of the penalty term. Intuitive values for α would be 1 or $1/2$ (and $\alpha = 0$ gets us back to standard case).

More precisely, the renormalized Lasso is parameterized by $\lambda \geq 0$ and chooses $\mathbf{u}_1 = (0, \ldots, 0)$ and, for $t \geq 2$,

$$\mathbf{u}_t(\lambda, \alpha) \in \operatorname*{argmin}_{\mathbf{u}\in\mathbb{R}^N} \left[\lambda \left(\sum_{t'=1}^{t-1} |\mathcal{N}_{t'}| \right)^{\alpha} \|\mathbf{u}\|_1 + \sum_{t'=1}^{t-1} \sum_{s\in\mathcal{N}_{t'}} \left(\mathbf{u}\cdot\mathbf{x}_{t'}^s - y_{t'}^s\right)^2 \right] \tag{3}$$

for the renormalized non-discounted version of the Lasso, and

$$\mathbf{u}_t(\lambda,\alpha;\beta,\gamma) \in \operatorname*{argmin}_{\mathbf{u}\in\mathbb{R}^N} \left[\lambda \left(\sum_{t'=1}^{t-1} \beta_{t-t'} \, |\mathcal{N}_{t'}| \right)^{\alpha} \|\mathbf{u}\|_1 + \sum_{t'=1}^{t-1} \beta_{t-t'} \sum_{s\in\mathcal{S}_{t'}} \left(\mathbf{u}\cdot\mathbf{x}_{t'}^s - y_{t'}^s\right)^2 \right] \quad (4)$$

for the renormalized discounted Lasso.

This variant of the (possibly discounted) Lasso makes the minimization-criterion defining the Lasso more homogeneous and enables to better customize the Lasso penalty with respect to the time round t. Consider the no-discount case for instance. Definition 3 is equivalent to

$$\mathbf{u}_t(\lambda,\alpha) \in \operatorname*{argmin}_{\mathbf{u}\in\mathbb{R}^N} \left[\lambda \left(\sum_{t'=1}^{t-1} |\mathcal{N}_{t'}| \right)^{\alpha-1} \|\mathbf{u}\|_1 \; + \; \frac{1}{\sum_{t'=1}^{t-1} |\mathcal{N}_{t'}|} \sum_{t'=1}^{t-1} \sum_{s\in\mathcal{N}_{t'}} \left(\mathbf{u}\cdot\mathbf{x}_{t'}^s - y_{t'}^s\right)^2 \right] \quad (5)$$

The second term of the right hand side of Equation 5 is an average (summand divided by number of terms in the summations). The choice of $\alpha = 1$ thus corresponds to a penalty of constant strength over time, that does not become negligible with time. That was not the case for the (possibly discounted version of the) Lasso (where $\alpha = 0$). Recent stochastic-type theoretical results very briefly recalled in Section 5.2 suggest to take $\alpha - 1$ of the order of $-1/2$, i.e., α of the order of $1/2$.

Finally, denote by $\lambda^*(\alpha)$ (resp. $\lambda^*(\alpha;\beta,\gamma)$) the parameter value λ minimizing the first-30-days-excluded RMSE of the renormalized Lasso (resp. renormalized discounted version of the Lasso) with parameter value α — assuming existence and uniqueness:

$$\lambda^*(\alpha) = \operatorname*{argmin}_{\lambda\geq 0} \mathrm{RMSE}_{t_0}\left(\mathbf{u}_0(\lambda,\alpha),\ldots,\mathbf{u}_T(\lambda,\alpha)\right)$$

for the renormalized Lasso, and

$$\lambda^*(\alpha;\beta,\gamma) = \operatorname*{argmin}_{\lambda\geq 0} \mathrm{RMSE}_{t_0}\left(\mathbf{u}_0(\lambda,\alpha;\beta,\gamma),\ldots,\mathbf{u}_T(\lambda,\alpha;\beta,\gamma)\right)$$

for the renormalized discounted version of the Lasso.

5.4 Default values of the parameters

Implemented with default values of the parameters

- $\lambda = 3$ and $\alpha = 0.5$ for the no-discount case;
- $\lambda = 60$, $\alpha = 0.5$, $\beta = 1.5$, and $\gamma = 150$ in case of discounted losses.

5.5 Implementation

The renormalization just affects the penalty term, and the LARS-Lasso algorithm enables to compute the whole Lasso path in both no-discount or discount cases. It thus just remains to change the point λ at which the linear interpolation is computed. More precisely, one has to change step (4) of the Lasso implementation in Section 3.5 by what follows.

(4) compute the linear interpolation given $\widetilde{\lambda} := \lambda \left(\sum_{t'=1}^{t-1} |\mathcal{N}_{t'}| \right)^{\alpha}$,

$$\mathbf{u}_{t+1} = \begin{cases} \dfrac{\widetilde{\lambda} - \lambda_{t+1,k+1}}{\lambda_{t+1,k} - \lambda_{t+1,k+1}} \left(\mathbf{u}_{t+1,k} - \mathbf{u}_{t+1,k+1} \right) & \text{if } \lambda_{t+1,k+1} \leq \widetilde{\lambda} < \lambda_{t+1,k}, \\ (0, \ldots, 0) & \text{if } \widetilde{\lambda} \geq \lambda_{t+1,0}. \end{cases}$$

for the renormalized non-discounted version of the Lasso, or

(4) compute the linear interpolation given $\widetilde{\lambda} := \lambda \left(\sum_{t'=1}^{t-1} \beta_{t-t'} |\mathcal{N}_{t'}| \right)^{\alpha}$,

$$\mathbf{u}_{t+1} = \begin{cases} \dfrac{\widetilde{\lambda} - \lambda_{t+1,k+1}}{\lambda_{t+1,k} - \lambda_{t+1,k+1}} \left(\mathbf{u}_{t+1,k} - \mathbf{u}_{t+1,k+1} \right) & \text{if } \lambda_{t+1,k+1} \leq \widetilde{\lambda} < \lambda_{t+1,k}, \\ (0, \ldots, 0) & \text{if } \widetilde{\lambda} \geq \lambda_{t+1,0}. \end{cases}$$

for the renormalized discounted version of the Lasso.

5.6 Performance versus variable selection: Non-discounted losses

Here are the RMSE of the renormalized non-discounted version of the Lasso for different values of the parameters α and λ.

α	0	0.5	1
$\lambda = 0$	20.79	20.79	20.76
$\lambda = 1$	20.79	20.77	21.54
$\lambda = 100$	20.77	21.35	22.40

Simulations on a (λ, α)-mesh suggest that the best RMSE approximately equals 20.76 and is obtained for the (a posteriori) choice of $\alpha^* = 0.5$ and $\lambda^* = \lambda^*(\alpha^*) \approx 2.82$ — see Figure 15.

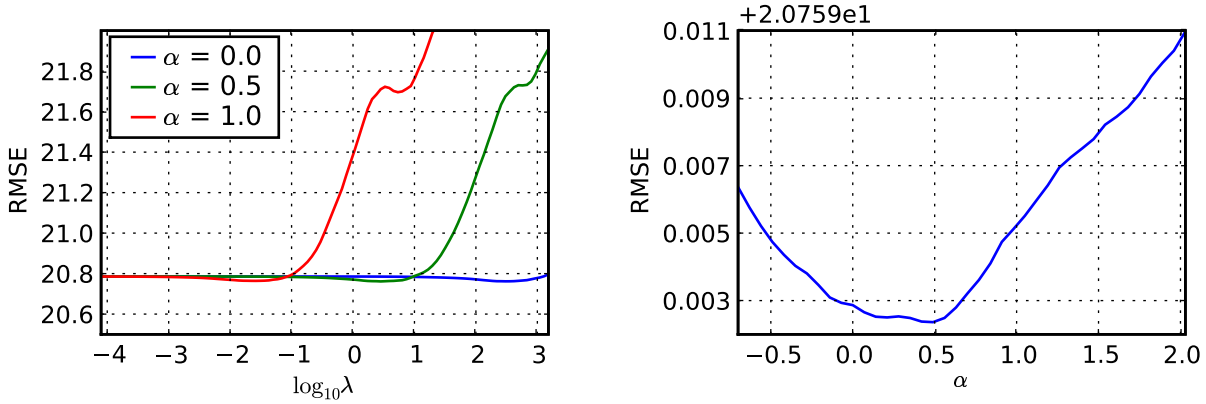

Figure 15: Practical performance with the N base forecasters versus λ for different values of the renormalization parameter α (left), or versus α with the corresponding optimal $\lambda = \lambda^*(\alpha)$ (right).

No significant gain in RMSE is thus enabled via renormalization (the best RMSE is just 5e-4 lower than that of the standard Lasso, i.e., 20.76 on the same data).

As far as variable selection is concerned, an evolution of the weights similar to the one for the standard Lasso can be observed — see Figure 16.

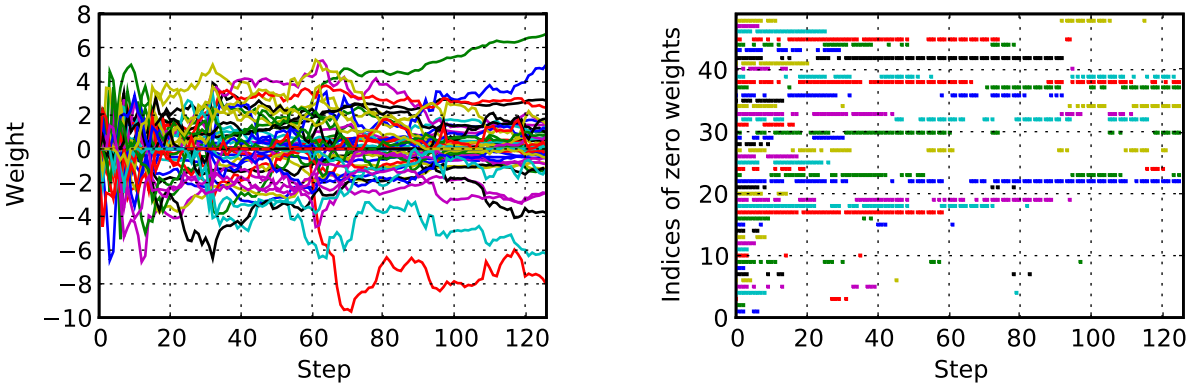

Figure 16: Evolution of the weights for $\alpha = \alpha^* \approx 0.5$ and $\lambda = \lambda^* \approx 2.81$ (left) ; corresponding zero weight indexes (right).

We first note the respective influence of the regularization parameter λ and the renormalization parameter α on the selectiveness of the renormalized Lasso. As shown on Figure 17, the larger the regularization parameter λ or the renormalization parameter α, the stronger the variable selection. This is again quite natural, since the penalty term is larger in both cases.

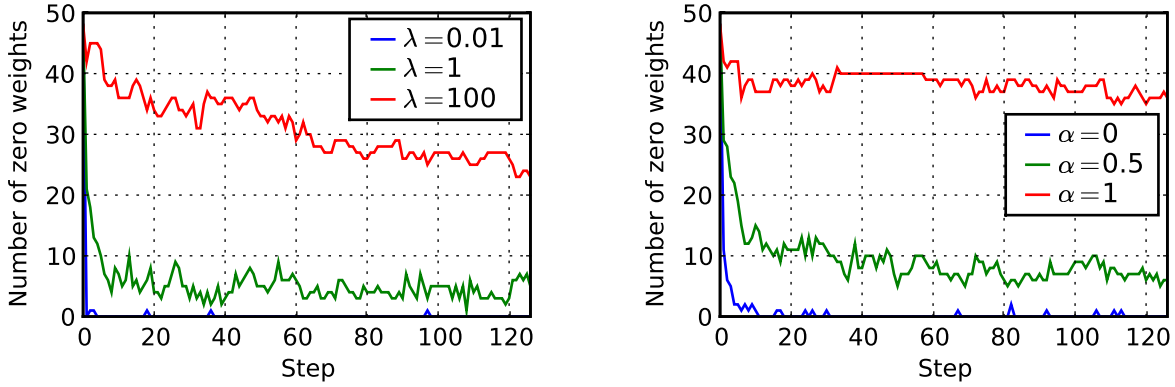

Figure 17: Evolution of the number of zero weights for $\alpha = \alpha^* \approx 0.5$ and different values of λ (left), and for $\lambda = \lambda^* \approx 2.82$ and different values of α (right).

The very interest of renormalizing w.r.t. variable selection lies in what could be expected intuitively: the selectivity of the renormalized version of the Lasso is more constant over time when α is closer to 1 (hence a better interpretability or an improved selection of a subset of the models), see Figure 18. A good compromise seems to be achieved with $\alpha^* = 0.5$: the renormalized version of the Lasso proves to be effective and well selective (at least more than with $\alpha = 0$).

Another crucial remark deals with calibration: by allowing the penalty term to depend on the total number of observations, the a posteriori optimal parameter λ may be used with the same kind of data but on different (longer, shorter) time periods.

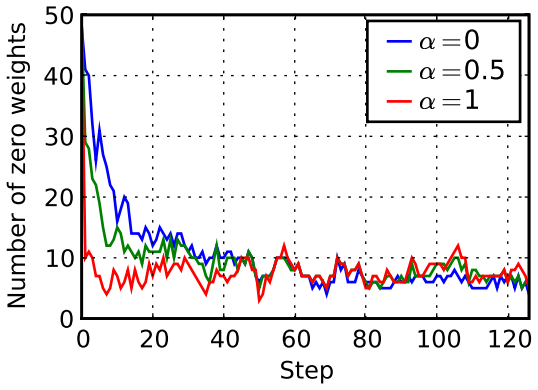

Figure 18: Evolution of the number of zero weights for different values of α and the corresponding values of $\lambda = \lambda^*(\alpha)$.

Penalty renormalization might also be useful to get a somehow universal regularization parameter λ, probably with the additional practical preprocessing that consists in renormalizing the data itself (reduction and centering of the observations and base predictions).

The same comments on robustness (slight under-penalization is suggested) and bounded parameter search for their calibration ((λ, α) here) hold as in the unnormalized case ($\alpha = 0$); see Sections 3.6 and 4.6.

5.7 Discounted losses

Simulations on a $(\lambda, \alpha, \beta, \gamma)$-mesh suggest that the best RMSE approximately equals 19.31 and is obtained for the (a posteriori) choice of $\beta^* \approx 1.5$, $\gamma^* \approx 150$, $\alpha^* = 0.5$ and $\lambda^* = \lambda^*(\alpha^*) \approx 60$.

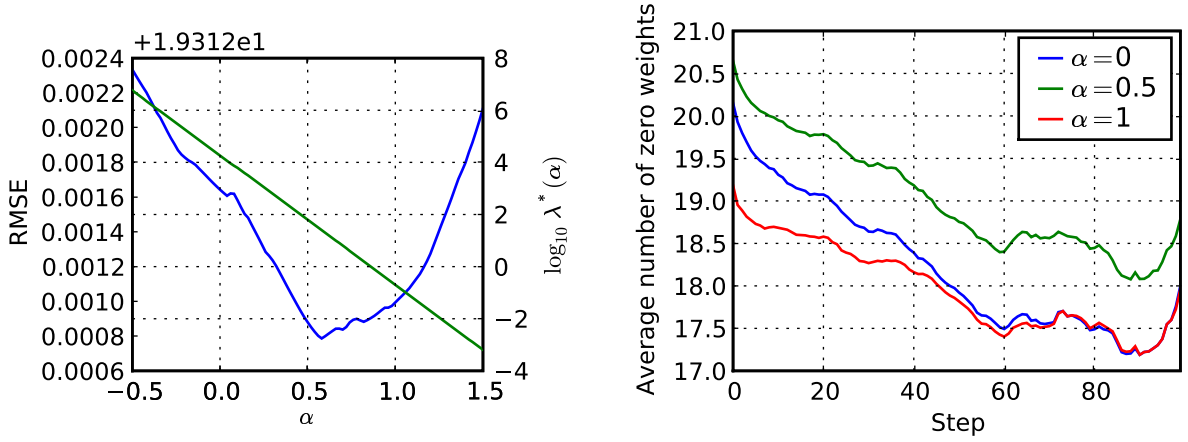

Figure 19: Practical performance (RMSE in blue) with the N base forecasters versus α with the corresponding optimal $\lambda = \lambda^*(\alpha)$ plotted in green (left); average number of zero weights after current step t for different values of α and the corresponding values of $\lambda = \lambda^*(\alpha)$ (right). In both cases, $\beta = 1.5$ and $\gamma = 150$.

No significant gain in RMSE stems from penalty renormalization (improvements are just of the order of 1.e-3: Figure 19, left).

Yet, Figure 19 shows that the renormalized discounted versions of the Lasso are highly selective. The one with $\alpha = 0.5$ is even among the most selective ones (right), since the average number of zero weights on the remaining time period $[t, T]$ is about one unit larger with $\alpha = 0.5$ than with $\alpha = 0$ (no renormalization) or $\alpha = 1$ (penalty of constant strength over time).

6 Calibration via Empirical Loss Minimization

6.1 References

This is a new calibration method. Tuning the parameters of learning algorithms has always been a delicate issue, and this is a new way out of it.

Some related practical attempts were made in stochastic scenarios to automatically choose some good parameter values (see, e.g., the empirical study based on cross-validation in [Gaïffas and Lecué, 2007]).

The previous theoretical results (see, e.g., [Cesa-Bianchi et al., 2007]), though totally adaptive and automatic, led to cautious updates of the weights and in turn, to poor performance, see [Mallet et al., 2007, Chapter 4].

6.2 Theoretical bound

None yet! The aim is to achieve a performance nearly as good as the one of the considered parameterized prediction method tuned with the best parameter in hindsight.

6.3 Interpretation and/or comments

This forecaster is a generic modification of all individual sequence prediction methods that depend on a (possibly vector-valued) tuning parameter. It automatically calibrates this parameter by choosing the parameter value that minimizes the past cumulative loss of the considered parameterized prediction method.

Several optimization methods might be used. We chose a grid-based optimization procedure, but continuous methods, though necessarily inaccurate and more difficult to control, might prove to be more effective. Good practical results were indeed obtained via the BFGS optimization procedure.

6.4 Statement and implementation

Consider a parameterized weighted average forecaster whose predictions at step t are computed with the weight vector $\mathbf{v}_t = \mathbf{v}_t^{(\lambda)}$ as described on page 3, where $\lambda \in \Lambda \subset \mathbf{R}^d$ is the parameter to be automatically chosen. For instance, $d = 1$ and $\Lambda = \mathbf{R}_+^*$ for the exponentiated gradient forecaster, the ridge regression forecaster and the Lasso.

We assume that $\mathbf{v}_1^{(\lambda)} = \mathbf{v}_1^*$ does not depend on λ, which is the case for every method studied so far.

Our forecaster automatically calibrated by minimization of the empirical loss (*ELM calibrated forecaster* thereafter) chooses $\mathbf{v}_1 = \mathbf{v}_1^*$ and, for $t \geq 2$,

$$\mathbf{v}_t = \mathbf{v}_t^{(\widehat{\lambda}_t)}$$

where

$$\widehat{\lambda}_t \in \operatorname*{argmin}_{\lambda\in\Lambda} \sum_{t'=1}^{t-1} \sum_{s\in\mathcal{N}_{t'}} \left(\mathbf{v}_{t'}^{(\lambda)} \cdot \mathbf{x}_{t'}^s - y_{t'}^s\right)^2 .$$

We chose a finite logarithmically-scaled parameter grid $\widetilde{\Lambda} \subset \Lambda$ to perform the above minimization.

Can be implemented as follows.

Parameters: the grid $\widetilde{\Lambda}$
Initialization: $\mathbf{v}_1 = \mathbf{v}_1^*$
For each round $t = 1, 2, \ldots, T$,

(1) predict with $\mathbf{v}_t$;

(2) get the observations and compute $\widehat{\lambda}_{t+1}$ as

$$\widehat{\lambda}_{t+1} \in \operatorname*{argmin}_{\lambda\in\widetilde{\Lambda}} \sum_{t'=1}^{t} \sum_{s\in\mathcal{N}_{t'}} \left(\mathbf{v}_{t'}^{(\lambda)} \cdot \mathbf{x}_{t'}^s - y_{t'}^s\right)^2 ;$$

(3) compute $\mathbf{v}_{t+1}$ as

$$\mathbf{v}_{t+1} = \mathbf{v}_{t+1}^{\left(\widehat{\lambda}_{t+1}\right)} .$$

Algorithmic remark: if the $\mathbf{v}_t^{(\lambda)}$ can be computed sequentially, one can save one order of computational (but not space) complexity in time step t by maintaining all weights $\mathbf{v}_t^{(\lambda)}$ (for all $\lambda \in \Lambda$).

6.5 Performance

We tested the calibration via empirical loss minimization on two simple prediction methods: the exponentiated gradient and the ridge regression forecasters.

We chose in both cases a uniform logarithmic grid:

– on [1e-8, 1e-4] for the exponentiated gradient;

– on [1, 1e6] for the ridge regression forecaster.

Below are summarized the first-30-days-excluded RMSE of the resulting ELM calibrated forecasters, for different numbers of grid points $|\widetilde{\Lambda}| = 11$, 101, 1001 and 10001.

These performances can be compared to the best RMSE obtained with a constant parameter λ ("best" — this is the target), or to the RMSE corresponding to a parameter value of 1e-8 (resp., 1e6) for the exponentiated gradient forecaster (resp., ridge regression forecaster) ("bad" — these particular parameter values yield the worse RMSE on the considered optimization grid when used at all time rounds).

$\lvert\widetilde{\Lambda}\rvert$	11	101	1001	10001	best	bad
EG	21.77	21.75	21.81	21.80	21.47	24.35
Ridge	20.81	20.81	20.81	20.81	20.77	21.80

We first note that, for both ridge regression and the exponentiated gradient forecaster, the grid does not need to be too dense. The obtained RMSE are indeed very stable as $|\widetilde{\Lambda}|$ increases, and a choice of $|\widetilde{\Lambda}| = 1000$ seems correct — the underlying forecasters are robust enough w.r.t. their parameters.

For both base forecasters, the RMSE of the associated ELM calibrated forecaster is not too far from the RMSE that could be obtained with the beforehand knowledge of the optimal choice of a constant parameter λ. The results are good, but could be better for the exponentiated gradient forecaster (a difference in the RMSE below 0.1 would be more acceptable). Yet, the achieved performance is much better than the one obtained with constant learning rate equal to 1e-8 (close to the ensemble mean).

As for the ridge regression forecaster, results are very good (one RMSE point below a bad one, and just 0.03 above the aimed one). Nevertheless, one should not forget that the ridge RMSE curve is very flat for small λ values, and then grows rapidly for λ large enough. The difficulty in calibrating the ridge regression forecaster thus just lies in avoiding too large λ (i.e., over-penalizing), which is not the case for the exponentiated gradient (large learning rates are bad, but small ones as well).

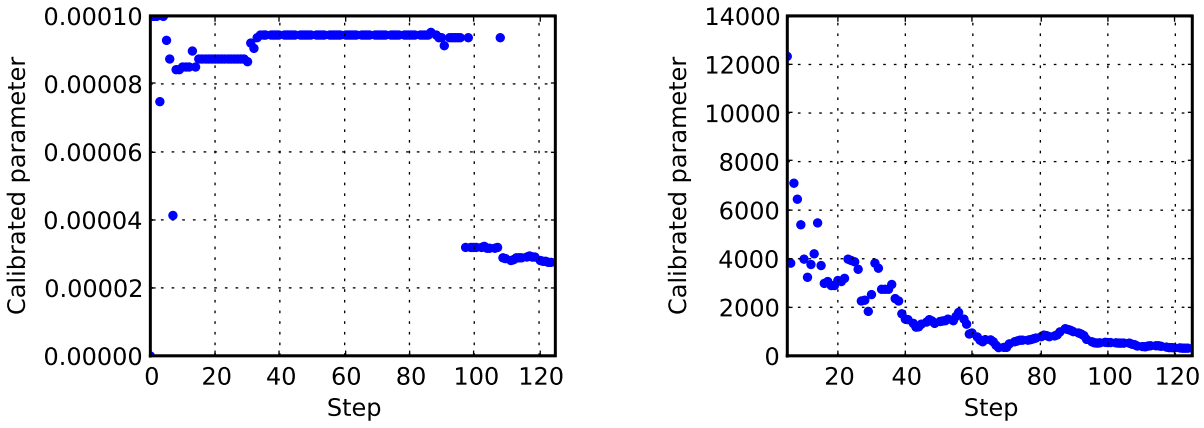

Figure 20: Evolution of the calibrated parameter $\widehat{\lambda}_t$ for the exponentiated gradient (left) and the ridge regression forecasters (first 5 days excluded because of too large parameter values, right). In both cases, $|\widetilde{\Lambda}| = 1001$.

Figure 20 shows the evolution of the calibrated parameters for these base forecasters. Note that the two main levels obtained for the exponentiated gradient forecaster are not artificial because the grid is dense enough (about 250 points per decade).

In a word, the numerical results presented above tend to prove that calibration via empirical loss minimization is quite performant.

6.6 Graphical evolution of the weights

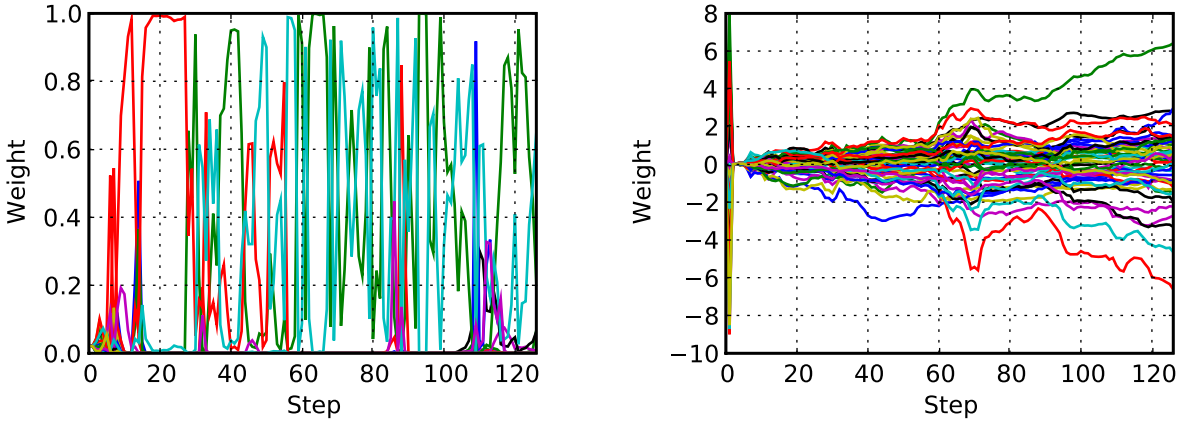

Figure 21: Evolution of the weights for the ELM calibrated exponentiated gradient (left) and the ELM calibrated ridge regression forecaster (right).

7 Data Renormalization

7.1 References

This consideration of some pre- and post-processing stages is a widespread practical technique in the statistics community, at least in stochastic settings — see
[Cornillon and Matzner-Lober, 2007] for instance. It is intuitively motivated by the i.i.d. case.

7.2 Theoretical bound

None yet! It is not even clear how to get one. An assumption of stochasticity is probably needed.

7.3 Interpretation and/or comments

This meta-forecaster is a black-box trick that can be applied to all individual sequence prediction methods, especially when they depend on some tuning parameters. It consists in a pre-treatment, which centers and reduces the base forecasters predictions and the observations before the weight vector is computed (the post-treatment counterpart follows).

The aim is threefold. It is first strongly motivated by the need to get a somehow universal parameter for parameterized prediction methods, e.g., independent of the range of the individual sequence

It might also be interesting for prediction methods minimizing a penalized M-criterion (penalized min-forecasters therafter), e.g., a cumulative square loss with a ℓ^1 or ℓ^2 penalty. If the N base forecasts are preprocessed, they then are treated more equally by the regularization.

Finally, centering the base predictions helps getting rid of the intercept term, and no constant term in the linear model seems intuitively necessary when the observations are centered too (hence a pure inner product $\mathbf{v}_t \cdot \mathbf{x}_t^s$). This can also be viewed as a way of partly eliminating the prediction biais.

7.4 Statement and implementation

We fix a weighted average forecaster $\mathcal{F}$ and will feed it with the following data.

For each $t \geq 2$ and each $m = 1, \ldots, N$, we define the empirical means and standard deviations of the past base predictions as

$$\overline{x}_{m,t-1} = \frac{1}{\sum_{t'=1}^{t-1} |\mathcal{N}_{t'}|} \sum_{t'=1}^{t-1} \sum_{s \in \mathcal{N}_{t'}} x_{m,t'}^s \ ,$$

and

$$\widehat{\sigma}_{m,t-1} = \left(\frac{1}{\sum_{t'=1}^{t-1} |\mathcal{N}_{t'}|} \sum_{t'=1}^{t-1} \sum_{s \in \mathcal{N}_{t'}} \left(x_{m,t'}^s - \overline{x}_{m,t-1} \right)^2 \right)^{\frac{1}{2}} \ .$$

We also define the empirical means and standard deviations of the past base observations as

$$\overline{y}_{t-1} = \frac{1}{\sum_{t'=1}^{t-1} |\mathcal{N}_{t'}|} \sum_{t'=1}^{t-1} \sum_{s \in \mathcal{N}_{t'}} y_{t'}^s \ ,$$

and

$$\widehat{\sigma^y}_{t-1} = \left(\frac{1}{\sum_{t'=1}^{t-1} |\mathcal{N}_{t'}|} \sum_{t'=1}^{t-1} \sum_{s \in \mathcal{N}_{t'}} \left(y_{t'}^s - \overline{y}_{t-1} \right)^2 \right)^{\frac{1}{2}} \ .$$

The meta-algorithm can be implemented as follows.

Parameters: A forecasting method $\mathcal{F}$ and its parameters

Initialization: As in $\mathcal{F}$; we denote by $\mathbf{v}_1$ the corresponding starting vector

For each round $t = 1, 2, \ldots, T$,

(1) if $t = 1$, predict $\widehat{y}_1^s = \mathbf{v}_1 \cdot \mathbf{x}_1^s$ at each station s; otherwise, predict

$$\widehat{y}_t^s = \overline{y}_{t-1} + \widehat{\sigma^y}_{t-1} \sum_{m=1}^{N} \mathbf{v}_{m,t} \frac{x_{m,t}^s - \overline{x}_{m,t-1}}{\widehat{\sigma}_{m,t-1}} \ ;$$

(2) get the observations and compute the quantities $\overline{x}_{m,t}$, $\widehat{\sigma}_{m,t}$, $m = 1, \ldots, N$, $\overline{y}_t$ and $\widehat{\sigma^y}_t$;

(3) compute the weight vector $\mathbf{v}_{t+1}$ prescribed by the forecasting method $\mathcal{F}$ when applied to the preprocessed past predictions and observations

$$\frac{\mathbf{x}_{m,t'}^s - \overline{x}_{m,t}}{\widehat{\sigma}_{m,t}} \ , \quad m = 1, \ldots, N \qquad \text{and} \qquad \frac{y_{t'}^s - \overline{y}_t}{\widehat{\sigma^y}_t} \ ,$$

for every $s \in \mathcal{N}_{t'}$ and $t' = 1, \ldots, t$.

Note that in the rare cases when the weight vector $\mathbf{v}_t$ computed by the forecasting method $\mathcal{F}$ depends on the base predictions at time round t, these base predictions should be preprocessed too before the computation of $\mathbf{v}_t$. This is for example the case for the modified non-linear version of the ridge regression forecaster — see [Cesa-Bianchi and Lugosi, 2006, page 320], or [Mallet et al., 2007, chapter 15].

7.5 Performance

We list below the results of some classical convex and linear prediction methods. They can be compared to the best RMSE obtained with a constant parameter on original data, without the pre- and post-processing steps ("best" thereafter, optimal value denoted with a star).

The discount parameters were taken equal to $\beta = 1.5$ and $\gamma = 150$ for both the ridge regression forecaster and the Lasso.

η	1e-3	0.01	0.03	0.1	1	best	η^*
EG	23.28	22.67	**22.62**	23.22	23.55	21.47	2e-5

λ	0.01	0.1	1	10	100	best	λ^*
Ridge	20.70	**20.69**	20.69	20.88	21.25	20.77	100
Ridge with discount	19.48	19.46	19.37	**19.26**	19.39	19.38	5.8e3
Lasso	20.70	20.69	**20.68**	20.91	21.64	20.76	300
Lasso with discount	19.48	19.48	19.44	**19.24**	19.32	19.31	2e4
Renormalized Lasso	**20.68**	21.05	21.74	22.91	35.07	20.76	3
Renormalized Lasso with discount	19.36	**19.17**	19.84	21.00	23.33	19.31	60

The best obtained RMSE are all slightly smaller than in the case with no pre-/post-processing of the data, except for the exponentiated gradient forecaster (because its combinations are convex).

Moreover, the corresponding optimal values for the parameters of the ridge regression forecaster and of the Lasso lie in a more reasonable scale; and such values could turn out to be more universal (provided pre-/post-processing steps are used). A sharper numerical study would yet be necessary to determine more precisely the optimal parameter value (we only reported the results of a sparse grid here).

8 Data Preprocessing

8.1 References and motivations

This meta-forecasting technique is inspired by the techniques of Section 7. It also applies to prediction methods minimizing a penalized M-criterion (penalized min-forecasters thereafter), e.g., a cumulative square loss with a ℓ^1 or ℓ^2 penalty, and enables to regularize the N base forecasters more equally.

It should not be applied to forecasting methods outputting convex combinations, only to the ones yielding linear combinations.

8.2 Theoretical bound

None yet! Theoretical results were proved in [de Geer, 2008] for the Lasso in the i.i.d. case.

8.3 Statement and implementation

We fix a weighted average forecaster $\mathcal{F}$ and will feed it with the following data.

For each $t \geq 2$ and each $m = 1, \ldots, N$, we define the empirical L^2 norms of the past base predictions as

$$\|\mathbf{x}_m\|_{t-1} = \left(\frac{1}{\sum_{t'=1}^{t-1} |\mathcal{N}_{t'}|} \sum_{t'=1}^{t-1} \sum_{s \in \mathcal{N}_{t'}} \left(x^s_{m,t'}\right)^2 \right)^{\frac{1}{2}} .$$

The meta-algorithm can be implemented as follows.

Parameters: A forecasting method $\mathcal{F}$ and its parameters

Initialization: As in $\mathcal{F}$; we denote by $\mathbf{v}_1$ the corresponding starting vector

For each round $t = 1, 2, \ldots, T$,

(1) if $t = 1$, predict $\widehat{y}^s_1 = \mathbf{v}_1 \cdot \mathbf{x}^s_1$ at each station s; otherwise, predict

$$\widehat{y}^s_t = \sum_{m=1}^{N} v_{m,t} \frac{x^s_{m,t}}{\|\mathbf{x}_m\|_{t-1}} ;$$

(2) get the observations and compute the quantities $\|\mathbf{x}_m\|_t$, $m = 1, \ldots, N$;

(3) compute the weight vector $\mathbf{v}_{t+1}$ prescribed by the forecasting method $\mathcal{F}$ when applied to the preprocessed past predictions

$$\frac{x^s_{m,t'}}{\|\mathbf{x}_m\|_t} , \quad m = 1, \ldots, N ,$$

for every $s \in \mathcal{N}_{t'}$ and $t' = 1, \ldots, t$.

Note that in the rare cases when the weight vector $\mathbf{v}_t$ computed by the forecasting method $\mathcal{F}$ depends on the base predictions at time round t, the same comments as after the statement of implementation in Section 7 apply.

8.4 Interpretation and/or comments

As mentioned in Section 8.1, this meta-method applied on penalized min-forecasters helps penalizing the base predictions in a more homogeneous way. This can be illustrated on the example of the ridge regression forecaster. For each time round $t \geq 2$, $\mathbf{v}_t$ is indeed defined as

$$\begin{aligned}
\mathbf{v}_t &= \operatorname*{argmin}_{\mathbf{u}\in\mathbb{R}^N} \left[\sum_{t'=1}^{t-1} \sum_{s\in\mathcal{S}_{t'}} \left(\sum_{m=1}^{N} u_m \frac{x^s_{t',m}}{\|\mathbf{x}_m\|_{t-1}} - y^s_{t'} \right)^2 + \lambda \sum_{m=1}^{N} u_m^2 \right] \\
&= \operatorname*{argmin}_{\mathbf{u}\in\mathbb{R}^N} \left[\sum_{t'=1}^{t-1} \sum_{s\in\mathcal{S}_{t'}} \left(\sum_{m=1}^{N} \frac{u_m}{\|\mathbf{x}_m\|_{t-1}} x^s_{t',m} - y^s_{t'} \right)^2 + \lambda \sum_{m=1}^{N} \|\mathbf{x}_m\|^2_{t-1} \left(\frac{u_m}{\|\mathbf{x}_m\|_{t-1}} \right)^2 \right] \\
&= \|\mathbf{x}_m\|_{t-1} \operatorname*{argmin}_{\mathbf{u}'\in\mathbb{R}^N} \left[\sum_{t'=1}^{t-1} \sum_{s\in\mathcal{S}_{t'}} \left(\sum_{m=1}^{N} u'_m x^s_{t',m} - y^s_{t'} \right)^2 + \lambda \sum_{m=1}^{N} \|\mathbf{x}_m\|^2_{t-1} u'^2_m \right] ,
\end{aligned}$$

where we used the change of variables

$$u'_m = \frac{u_m}{\|\mathbf{x}_m\|_{t-1}}, \quad m = 1, \ldots, N .$$

Base forecasters with larger empirical Euclidian norms are thus more penalized.

8.5 Performance

We list below the results of some classical linear prediction methods. They can be compared to the best RMSE obtained with a constant parameter on original data, without the pre-processing step ("best" thereafter, optimal value denoted with a star).

The discount parameters were taken equal to $\beta = 1.5$ and $\gamma = 150$ for both the ridge regression forecaster and the Lasso.

λ	0.01	0.1	1	10	100	best	λ^*
Ridge	**20.77**	20.83	21.12	21.49	21.81	20.77	100
Ridge with discount	19.56	19.44	**19.39**	19.67	20.17	19.38	5.8e3
Lasso	20.79	20.78	**20.77**	20.78	21.17	20.76	300
Lasso with discount	19.61	19.61	19.60	19.54	**19.33**	19.31	2e4
Renormalized Lasso	**20.77**	20.80	21.34	22.00	22.28	20.76	3
Renormalized Lasso with discount	19.59	19.44	**19.34**	20.28	20.89	19.31	60

There is no improvement in the RMSE, and even a slight worsening of the results with respect to the case with no pre-processing of the data. They are thus slighly larger than in the case of centering and reduction of the past base predictions and observations — see Section 7.

On the other hand, while the optimal value for the parameter of the ridge regression forecaster can be expected to be universal w.r.t. the range of the individual sequence, because of the ℓ^2 penalty is consistent with the square loss, this is not the case for the Lasso and its variants.

9 Selection of a Good Subset of the models by the Lasso

9.1 Motivations and comments

This aims at reducing the overall computational load by removing some less necessary models, hence shortening the simulation stage. For the time being, we provide such a study in hindsight only (i.e., run a first time the Lasso on the data, make a selection, and then run all the algorithms again on the same data feeding them with the selected subsample only).

This is a black-box trick that can be used preliminary to the consideration of any forecaster.

9.2 Basic ideas

We sketch below the basic ideas underlying the Lasso preselection. More sophisticated (e.g., adaptative) schemes could advantageously be implemented.

The Lasso preselection consists in first applying the Lasso to the whole ensemble on a given period, and then removing the models whose weights were most often set to zero by the Lasso. The remaining models are together called the "Lasso-preselected ensemble". The same can be done with the discounted version of the Lasso, yielding the "discounted-Lasso-preselected ensemble".

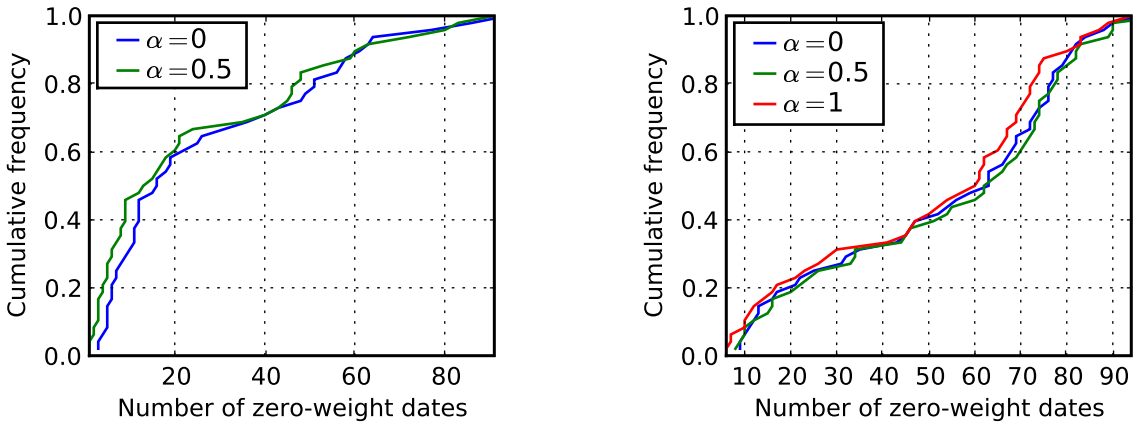

Figure 22: Distribution of the number of zero-weight dates over all the 48 initial models. The weights are computed by the non-discounted version of the Lasso (left), or by its discounted version (right).

To choose the subset of models to remove, we used a graphical procedure. We considered the distribution of the number of zero-weight dates (on the usual 126-day period) over the whole 48-member ensemble. Figure 22, left, plots the empirical cumulative distribution of the list of 48 elements, in which each element is the number of times a given model gets zero weight from the Lasso. It shows that the first 60% models most often kept by the Lasso cannot be easily distinguished (small variations in the number of zero-weight

dates), whereas the remaining 40% were left out a significantly larger number of times. We thus fixed the preselection level at 22 dates: only the models left out by the Lasso a number of times less or equal to 22 make up the preselected ensemble. This way, we removed 19 models (40%) among the 48 initial ones.

Since the distribution associated to the discounted version of the Lasso was almost uniform, no evident preselection level could be decided. We thus chose to keep the first 60% most Lasso-selected models as in the no-discount case to compute the discounted-Lasso-preselected ensemble (hence a preselection level at 68 dates).

9.3 Performance

We fixed the Lasso parameters to their optimal values (in hindsight) for the whole period and the whole set of models to get a first run of these forecasters. We then computed the Lasso-preselected and discounted-Lasso-preselected ensembles. We finally applied the following prediction methods with their previous (in hindsight) optimal parameters, to the whole test period but using only the subsamples (and computing the associated RMSE by discarding the first 30 days as a learning period, as usual).

A more sophisticated test (dissociating the preselection and prediction periods) would certainly be preferable.

Preselection with	Lasso	Disc. Lasso	Previous RMSE	for $\eta =$
EG	21.86	21.42	21.47	2e-5

Preselection with	Lasso	Disc. Lasso	Previous RMSE	for $\lambda =$
Ridge	20.77	20.83	20.77	100
Ridge with discount	19.39	19.36	19.38	5.8e3
Lasso	20.77	20.82	20.76	300
Lasso with discount	19.43	19.31	19.31	2e4

Results are positive for both the ridge regression forecaster and the Lasso, since similar RMSE are obtained with 29 models instead of 48 in both cases.

10 Results on a larger ensemble and on a longer period

Some usual convex or linear prediction methods were tested on a larger ensemble with 100 members and on a longer period [01/01/2001 – 30/12/2001], including the shorter reference period considered before (summer 2001). Note that the former 48 models are not included (even in the summer period) in this larger ensemble. The RMSE are as usual computed out of the short initial learning period of 30 days.

The smallest RMSE obtained with a constant parameter can be compared to the one of the best model: "Gain with 100" stands for the corresponding improvement (i.e., difference between the two RMSE) on the new, larger, ensemble. "Former gain" is the improvement made w.r.t. the (best of the) 48-member ensemble used in the rest of this report on the shorter period. We mostly indicate it to show that the interest of aggregation techniques increases with time.

The best model out of the 100 ones has a RMSE that equals 22.67 on the longer period considered. We also indicate three oracles computed on the ensemble with 100 members: the smallest RMSE that can be achieved with a constant convex, respectively, linear, combination equals 22.21, respectively, 20.00. The smallest RMSE that can be achieved with a time-varying linear combination (the "prescient") finally equals 7.53.

10.1 Performance without data renormalization

Method (secondary parameters)	Best RMSE	Opt. parameter	Gain with 100	Former gain
EG	20.87	1.8e-5	1.81	0.96
Thresholded EG ($\gamma = 1/100$)	20.89	1.7e-5	1.78	0.96
Ridge	20.64	2.1e4	2.03	1.66
Ridge with discount ($\beta = 1.5$ and $\gamma = 150$)	**19.35**	2.5e5	3.32	3.05
Lasso	20.69	4200	1.98	1.67
Lasso with discount ($\beta = 1.5$ and $\gamma = 150$)	19.41	4.5e4	3.26	3.12
Renormalized Lasso ($\alpha = 0.5$)	20.69	17	1.98	1.67
Renormalized Lasso with discount ($\alpha = 0.5$, $\beta = 1.5$ and $\gamma = 150$)	19.42	190	3.26	3.12

The linear oracle is now clearly outperformed.

10.2 Performance with data renormalization

Data renormalization is considered as described in Section 7. Forecasting is carried out with the tuning parameters recommended after the study therein. Note that a further numerical study would be necessary to determine the optimal parameter values more precisely (with a finer grid, both here and in Section 7).

Method (secondary parameters)	Best RMSE	Opt. parameter	Gain with 100	Former gain
EG	22.69	8.5e-3	-0.02	-0.19
Ridge	20.51	100	2.16	1.74
Ridge with discount ($\beta = 1.5$ and $\gamma = 150$)	**19.28**	1000	3.39	3.17
Lasso	20.55	10	2.12	1.75
Lasso with discount ($\beta = 1.5$ and $\gamma = 150$)	19.32	100	2.12	3.19
Renormalized Lasso ($\alpha = 0.5$)	20.59	0.01	2.08	1.75
Renormalized Lasso with discount ($\alpha = 0.5$, $\beta = 1.5$ and $\gamma = 150$)	19.37	1	3.30	3.26

10.3 Performance of the ELM calibration algorithm

The same experiment configuration as in Section 6.5 was chosen.

$\lvert\widetilde{\Lambda}\rvert$	11	101	1001	best	bad
EG	20.90	20.89	20.89	20.87	24.59
Ridge	20.66	20.66	20.66	20.64	21.10

ELM calibration thus proves to be very effective on the 100-member ensemble.

We can note on Figure 23 that the ELM calibrated exponentiated gradient parameter tends to be constant in the limit. As for the ELM ridge regression forecaster, its parameter can be shown to be approximately linear on the second half period (a penalty renormalization as with the Lasso could thus help to get the right optimal penalty evolution without the ELM procedure). Note also a sudden and temporary peak of this parameter at about $t = 50$, which has not been explained yet.

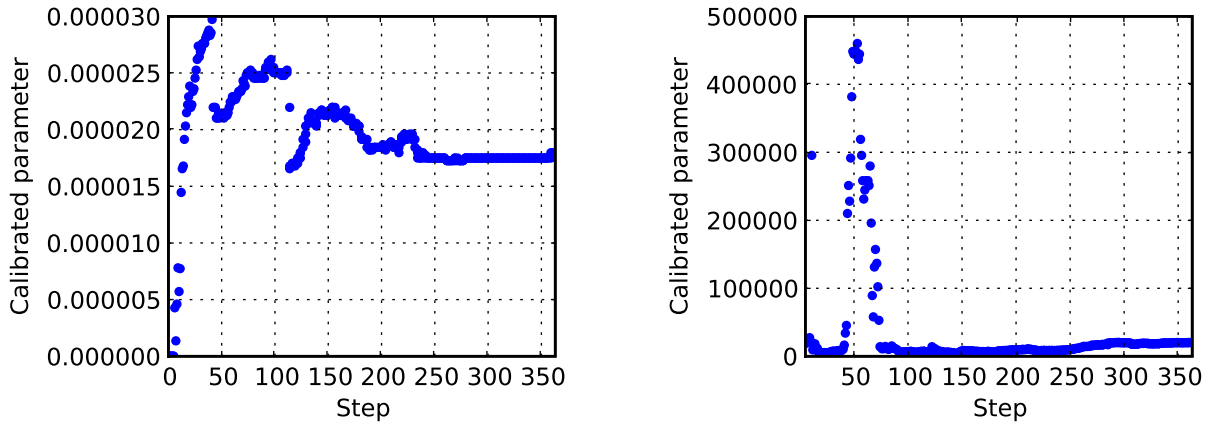

Figure 23: Evolution of the calibrated parameter $\widehat{\lambda}_t$ for the exponentiated gradient (left) and the ridge regression forecasters. In both cases, $\lvert\widetilde{\Lambda}\rvert = 1001$.

References

[Allenberg et al., 2006] Allenberg, C., Auer, P., Györfi, L., and Ottucsak, G. (2006). Hannan consistency in on-line learning in case of unbounded losses under partial monitoring. In *Proceedings of ALT'2006*, pages 229–243. Springer.

[Bickel et al., 2008] Bickel, P. J., Ritov, Y., and Tsybakov, A. (2008). Simultaneous analysis of lasso and dantzig selector. *Ann. Stat.* To appear.

[Bunea et al., 2007] Bunea, F., Tsybakov, A., and Wegkamp, M. (2007). Sparsity oracle inequalities for the lasso. *Elec. J. Stat.*, 1:169–194.

[Cesa-Bianchi and Lugosi, 2006] Cesa-Bianchi, N. and Lugosi, G. (2006). *Prediction, learning, and games*. Cambridge University Press.

[Cesa-Bianchi et al., 2007] Cesa-Bianchi, N., Mansour, Y., and Stoltz, G. (2007). Improved second-order bounds for prediction with expert advice. *Mach. Learn.*, 66(2/3):321–352.

[Cornillon and Matzner-Lober, 2007] Cornillon, P.-A. and Matzner-Lober, E. (2007). *Régression, Théorie et applications*. Springer.

[de Geer, 2008] de Geer, S. A. V. (2008). High-dimensional generalized linear models and the lasso. *Ann. Stat.*, 36(2):614–645.

[Efron et al., 2004] Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *Ann. Stat.*, 32(2):407–499.

[Gaïffas and Lecué, 2007] Gaïffas, S. and Lecué, G. (2007). Optimal rates and adaptation in the single-index model using aggregation. *Elec. J. Stat.*, 1:538–573.

[Mallet et al., 2007] Mallet, V., Mauricette, B., and Stoltz, G. (2007). Description of sequential aggregation methods and their performances for ozone ensemble forecasting. DMA-07-08 2007a, École Normale Supérieure, Paris.

[Osborne et al., 2000a] Osborne, M. R., Presnell, B., and Turlach, B. A. (2000a). A new approach to variable selection in least squares problems. *IMA J. Num. Anal.*, 20:389–404.

[Osborne et al., 2000b] Osborne, M. R., Presnell, B., and Turlach, B. A. (2000b). On the lasso and its dual. *J. Comp. Graph. Stat.*, 9(2):319–337.

[Tibshirani, 1996] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. R. Statist. Soc.*, 58(1):267–288.