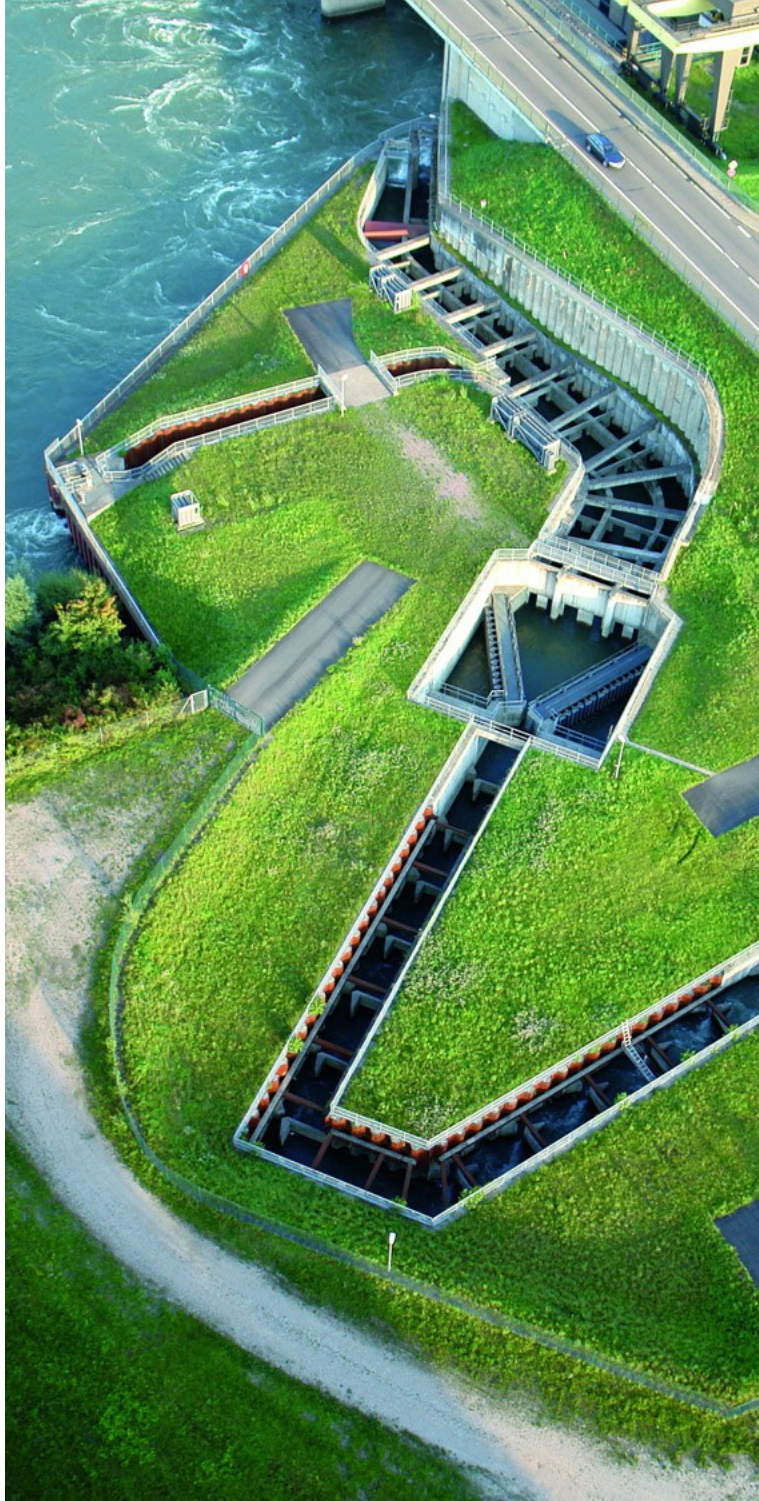
Forecasting the electricity consumption by aggregating specialized experts

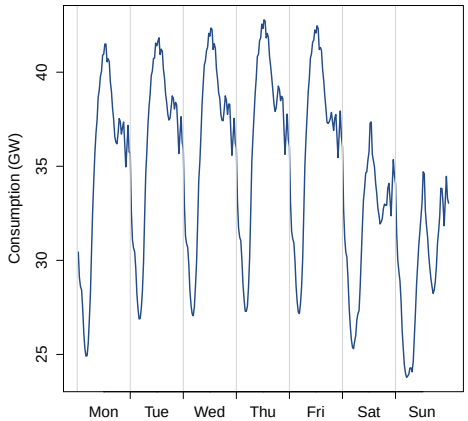
Pierre Gaillard (EDF R&D, ENS Paris)

with Yannig Goude (EDF R&D)
Gilles Stoltz (CNRS, ENS Paris, HEC Paris)

June 2013 – STAT



Goal



Goal

Short-term (one-day-ahead) forecasting of the French electricity consumption

Many models developed by EDF R&D: parametric, semi-parametric, and non-parametric

Evolution of the electrical scene in France
$\Rightarrow$ existing models get questionable

} Adaptive methods of models aggregation

Setting – Sequential prediction with expert advice

Each instance t

- Each expert suggests a prediction $x_{i,t}$ of the consumption y_t
- We assign weight to each expert and we predict

$$\widehat{y}_t = \widehat{\boldsymbol{p}}_t \cdot \boldsymbol{x}_t \quad \left(= \sum_{i=1}^{N} \widehat{p}_{i,t} x_{i,t} \right)$$

Our goal is to minimize our cumulative loss

$$\underbrace{\sum_{t=1}^{T} (\widehat{y}_t - y_t)^2}_{\text{Our loss}} = \underbrace{\min_{i=1,\dots,N} \sum_{t=1}^{T} (x_{i,t} - y_t)^2}_{\text{Loss of the best expert}} + \underbrace{R_T}_{\text{Estimation error}}$$

Good set of experts

Good aggregating algorithm

Setting – Sequential prediction with expert advice

Each instance t

- Each expert suggests a prediction $x_{i,t}$ of the consumption y_t
- We assign weight to each expert and we predict

$$\widehat{y}_t = \widehat{\boldsymbol{p}}_t \cdot \boldsymbol{x}_t \quad \left(= \sum_{i=1}^{N} \widehat{p}_{i,t} x_{i,t} \right)$$

Our goal is to minimize our cumulative loss

$$\underbrace{\sum_{t=1}^{T} (\widehat{y}_t - y_t)^2}_{\text{Our loss}} = \underbrace{\min_{\boldsymbol{q} \in \Delta_N} \sum_{t=1}^{T} (\boldsymbol{q} \cdot \boldsymbol{x}_t - y_t)^2}_{\text{Loss of the best convex combination}} + \underbrace{R_T}_{\text{Estimation error}}$$

Good set of experts
As varied as possible

Good aggregating algorithm

Minimizing both approximation and estimation error

$$\underbrace{\sum_{t=1}^{T}(\widehat{y}_t - y_t)^2}_{\text{Our loss}} = \underbrace{\min_{\boldsymbol{q}\in\Delta_N} \sum_{t=1}^{T}(\boldsymbol{q}\cdot\boldsymbol{x}_t - y_t)^2}_{\text{Approximation error}} + \underbrace{R_T}_{\text{Estimation error}}$$

Approximation error

$\Rightarrow$ good heterogeneous set of experts

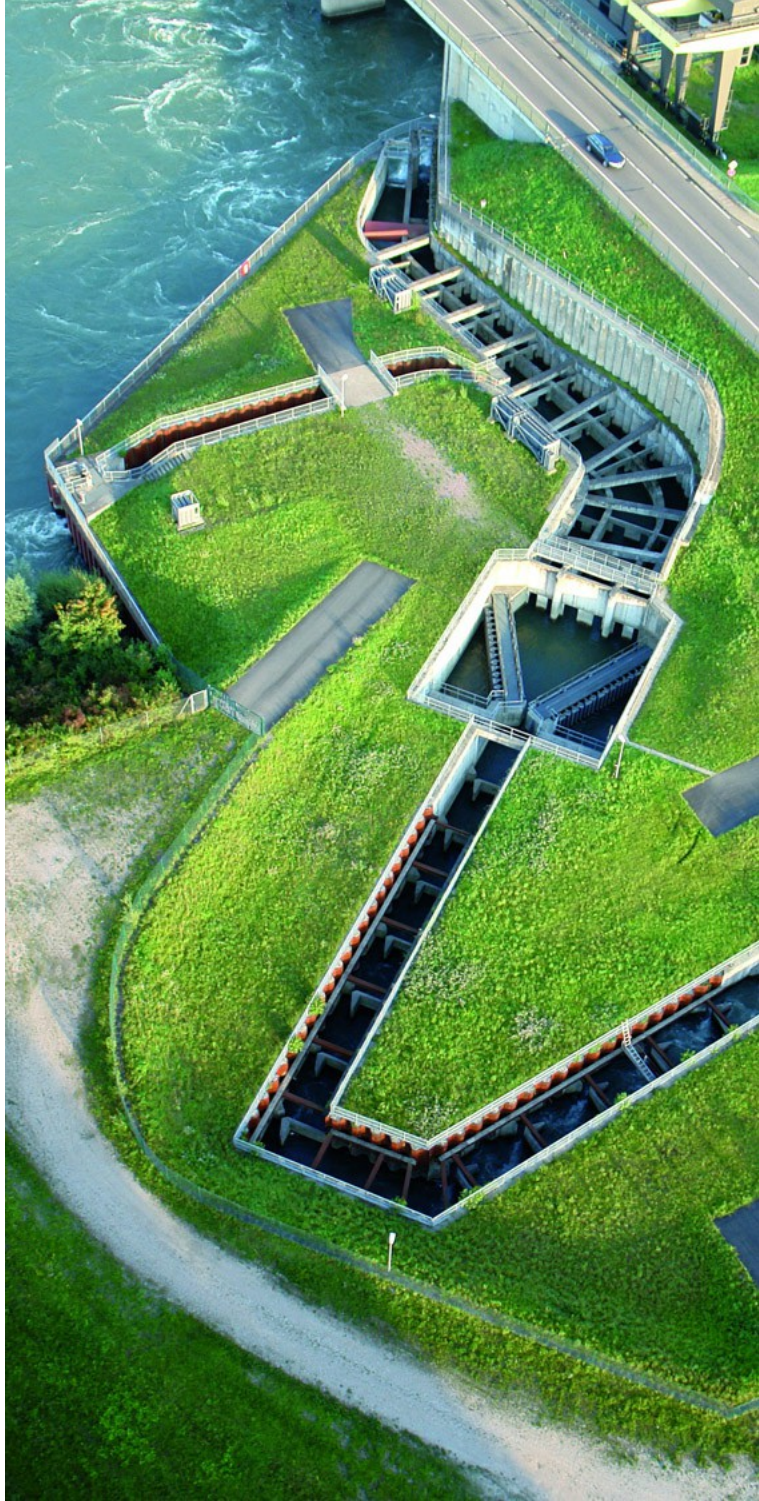
Ex: specializing the experts, bagging, boosting, ...

Estimation error

$\Rightarrow$ efficient algorithm for aggregating specialized experts

Ex: Exponentially weighted average, Exponentiated Gradient, Ridge, ...

Prediction Learning and Games, Cesa-Bianchi and Lugosi, 2006



I. Aggregating algorithms

Prediction Learning and Games, Cesa-Bianchi and Lugosi, 2006

June 2013 – STAT

Exponentially weighted average forecaster (EWA)

Each instance t

- Each expert suggests a prediction $x_{i,t}$ of the consumption y_t
- We assign to expert i the weight

$$\widehat{p}_{i,t} = \frac{exp\left(-\eta \sum_{s=1}^{t-1}(x_{i,s} - y_s)^2\right)}{\sum_{j=1}^{N} exp\left(-\eta \sum_{s=1}^{t-1}(x_{j,s} - y_s)^2\right)}$$

- and we predict $\widehat{y}_t = \sum_{i=1}^{N} \widehat{p}_{i,t} x_{i,t}$

Our cumulated loss is upper bounded by

$$\underbrace{\sum_{t=1}^{T}(\widehat{y}_t - y_t)^2}_{\text{Our loss}} \leqslant \underbrace{\min_{i=1,\dots,d} \sum_{t=1}^{T}(x_{i,t} - y_t)^2}_{\text{Loss of the best expert}} + \underbrace{\square\sqrt{T \log N}}_{\text{Estimation error}}$$

Proof

Lemme (Hoeffding)

Let X be a random variable taking values in $[a, b]$. Then for any $s \in \mathbb{R}$,

$$\ln \mathbb{E}\left[e^{sX}\right] \leqslant s\mathbb{E}[X] + \frac{s^2(b-a)}{8}$$

1. Upper bound the instantaneous loss $(\widehat{x}_t - y_t)^2$

$$\begin{aligned}
\left(\widehat{\boldsymbol{p}}_t \cdot \boldsymbol{x}_t - y_t\right)^2 &\overset{\text{by convexity}}{\leqslant} \widehat{\boldsymbol{p}}_t \cdot (\boldsymbol{x}_t - y_t)^2 \\
&\overset{\text{by Hoeffding}}{\leqslant} -\frac{1}{\eta} \ln \left(\sum_{j=1}^{d} \widehat{p}_{j,t} e^{-\eta (x_{j,t} - y_t)^2} \right) + \frac{\eta}{8} \\
&= -\frac{1}{\eta} \ln \left(\frac{\widehat{p}_{i,t}}{\widehat{p}_{i,t+1}} e^{-\eta (x_{i,t} - y_t)^2} \right) + \frac{\eta}{8} \\
&= (x_{i,t} - y_t)^2 + \frac{1}{\eta} \ln \frac{\widehat{p}_{i,t+1}}{\widehat{p}_{i,t}} + \frac{\eta}{8}
\end{aligned}$$

2. Summing over all t and telescoping

$$\sum_{t=1}^{T} (\widehat{x}_t - y_t)^2 - (x_{i,t} - y_t)^2 \leqslant \frac{1}{\eta} \ln \frac{\widehat{p}_{i,T+1}}{\widehat{p}_{i,1}} + \frac{\eta T}{8} = \sqrt{\frac{T}{8} \ln N} \quad \text{for } \eta = \sqrt{8 \ln N / T}$$

Exponentially weighted average forecaster (EWA)

Each instance t

- Each expert suggests a prediction $x_{i,t}$ of the consumption y_t
- We assign to expert i the weight

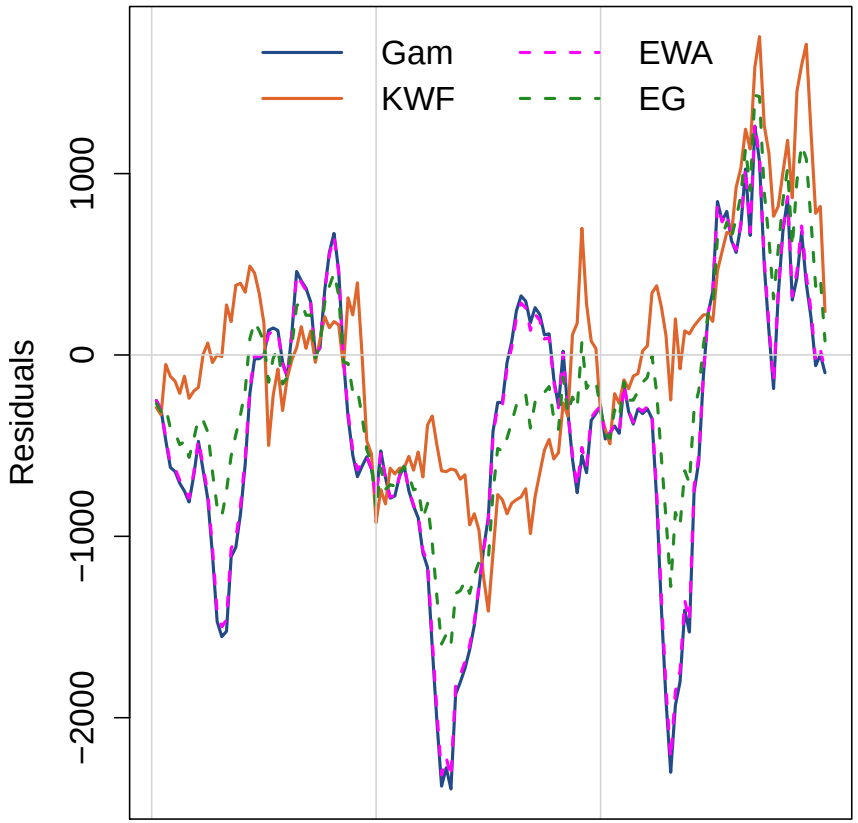
$$\widehat{p}_{i,t} = \frac{exp\left(-\eta \sum_{s=1}^{t-1}(x_{i,s} - y_s)^2\right)}{\sum_{j=1}^{N} exp\left(-\eta \sum_{s=1}^{t-1}(x_{j,s} - y_s)^2\right)}$$

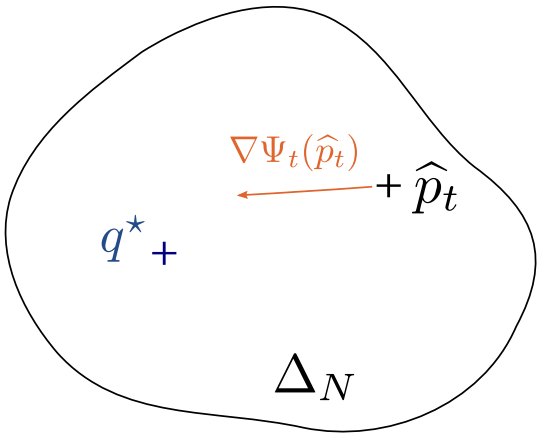
- and we predict $\widehat{y}_t = \sum_{i=1}^{N} \widehat{p}_{i,t} x_{i,t}$

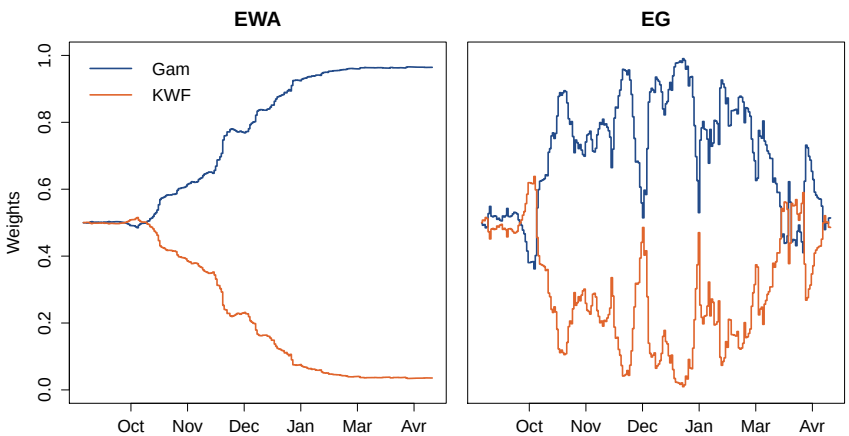
Our cumulated loss is upper bounded by

$$\underbrace{\sum_{t=1}^{T}(\widehat{y}_t - y_t)^2}_{\text{Our loss}} \leq \underbrace{\min_{q \in \Delta_N} \sum_{t=1}^{T}(q \cdot x_t - y_t)^2}_{\text{Loss of the best convex combination}} + \underbrace{?}_{\text{Estimation error}}$$

Motivation of convex combinations







Exponentiated gradient forecaster (EG)

Each instance t

- Each expert suggests a prediction $x_{i,t}$ of the consumption y_t
- We assign to expert i the weight

$$\widehat{p}_{i,t} \propto exp\left(-\eta \sum_{s=1}^{t-1} \ell_{i,s}\right) \qquad \text{where } \ell_{i,s} = 2(\widehat{y}_s - y_s)x_{i,s}$$

- and we predict $\widehat{y}_t = \sum_{i=1}^{N} \widehat{p}_{i,t} x_{i,t}$

Our cumulated loss is then bounded as follow

$$\underbrace{\sum_{t=1}^{T} (\widehat{y}_t - y_t)^2}_{\text{Our loss}} \leqslant \underbrace{\min_{\boldsymbol{q} \in \Delta_N} \sum_{t=1}^{T} (\boldsymbol{q} \cdot \boldsymbol{x}_t - y_t)^2}_{\text{Loss of the best convexe combination}} + \underbrace{\square \sqrt{T \log N}}_{\text{Estimation error}}$$

Idea of proof

$$\begin{aligned} \sum_{t=1}^{T} (\widehat{y}_t - y_t)^2 - (\boldsymbol{q}^\star \cdot \boldsymbol{x}_t - y_t)^2 &\leqslant \sum_{t=1}^{T} \underbrace{2(\widehat{\boldsymbol{p}}_t \cdot \boldsymbol{x}_t - y_t)\boldsymbol{x}_t}_{\ell_t} \cdot (\widehat{\boldsymbol{p}}_t - \boldsymbol{q}^\star) \\ &= \sum_{t=1}^{T} \ell_t \cdot (\widehat{\boldsymbol{p}}_t - \boldsymbol{q}^\star) \\ &\leqslant \sum_{t=1}^{T} \widehat{\boldsymbol{p}}_t \cdot \ell_t - \min_i \sum_{t=1}^{T} \ell_{i,t} \end{aligned}$$

Online tuning of the learning parameter η

The optimal value of the parameter

$$\eta^\star = \square\sqrt{\frac{\ln N}{T}}$$

is not necessarly known in advance if we do not know the horizon T.
Hence η has to be online calibrated.

Theoretical way. Use time varying parameters $\widehat{\eta}_t = \sqrt{\frac{\ln N}{t}}$

Practical way. Consider a grid Λ of potential parameters

`Initialize:` $\Lambda = \{1\}$

At each instance t

- Choose $\widehat{\eta}_t$ the best parameter in Λ so far
- If it is on a border, increase Λ exponentially

Fast rate?

The results stated above still stand for any bounded loss function convex in our prediction $\ell : (x, y) \mapsto \ell(x, y)$

$$\sum_{t=1}^{T} \ell(\widehat{y}_t, y_t) \quad \leqslant \quad \min_{\boldsymbol{q} \in \Delta_N} \quad \sum_{t=1}^{T} \ell(\boldsymbol{q} \cdot \boldsymbol{x}_t, y_t) \quad + \square \sqrt{T \log N}$$

If the loss function ℓ is η-exp-concave ie. $x \mapsto e^{-\eta \ell(x,y)}$ is concave for all y then by computing a weighted average on the whole simplex

$$\widehat{p}_t \propto \int_{\Delta_N} \boldsymbol{q} e^{-\eta \sum_{s=1}^{t-1} \ell(\boldsymbol{q} \cdot \boldsymbol{x}_t, y_t)} d\mu(\boldsymbol{q})$$

one can get

$$\sum_{t=1}^{T} \ell(\widehat{y}_t, y_t) \quad \leqslant \quad \min_{\boldsymbol{q} \in \Delta_N} \quad \sum_{t=1}^{T} \ell(\boldsymbol{q} \cdot \boldsymbol{x}_t, y_t) \quad + \square \frac{N \ln T}{\eta}$$

Example. The square loss $x \mapsto (x - y)^2$ is $1/2$-exp-concave on $[0, 1]^2$.

Other notions of regret?

Shifting regret

$$\sum_{t=1}^{T} \ell(\widehat{y}_t, y_t) \quad \leqslant \quad \min_{\substack{\boldsymbol{q}_1, \ldots, \boldsymbol{q}_T \\ \text{st } [\ldots] \leqslant m}} \quad \sum_{t=1}^{T} \ell(\boldsymbol{q}_t \cdot \boldsymbol{x}_t, y_t) \quad + \quad \square \sqrt{mT \ln \frac{NT}{m}}$$

Adaptive regret

$$\max_{s-r+1 \leqslant \tau_T} \left\{ \sum_{t=r}^{s} \ell(\widehat{y}_t, y_t) \quad - \quad \min_{\boldsymbol{q} \in \Delta_d} \quad \sum_{t=r}^{s} \ell(\boldsymbol{q} \cdot \boldsymbol{x}_t, y_t) \right\} \leqslant \quad \square \sqrt{\tau_T \ln(N\tau_T)}$$

Discounted regret

$$\max_{\sum_t \gamma_t \leqslant \mathcal{T}_T} \left\{ \sum_{t=1}^{T} \gamma_t \ell(\widehat{y}_t, y_t) \quad - \quad \min_{\boldsymbol{q} \in \Delta_d} \quad \sum_{t=1}^{T} \gamma_t \ell(\boldsymbol{q} \cdot \boldsymbol{x}_t, y_t) \right\} \leqslant \quad \square \sqrt{\mathcal{T}_T \ln(N\mathcal{T}_T)}$$

Cesa-Bianchi, Gaillard, Lugosi, and Stoltz, NIPS 2012

Fixed-Share algorithm, Herbster & Warmuth 1998

Each instance t

- Each expert suggests a prediction $x_{i,t}$ of the consumption y_t
- We assign to expert i the weight

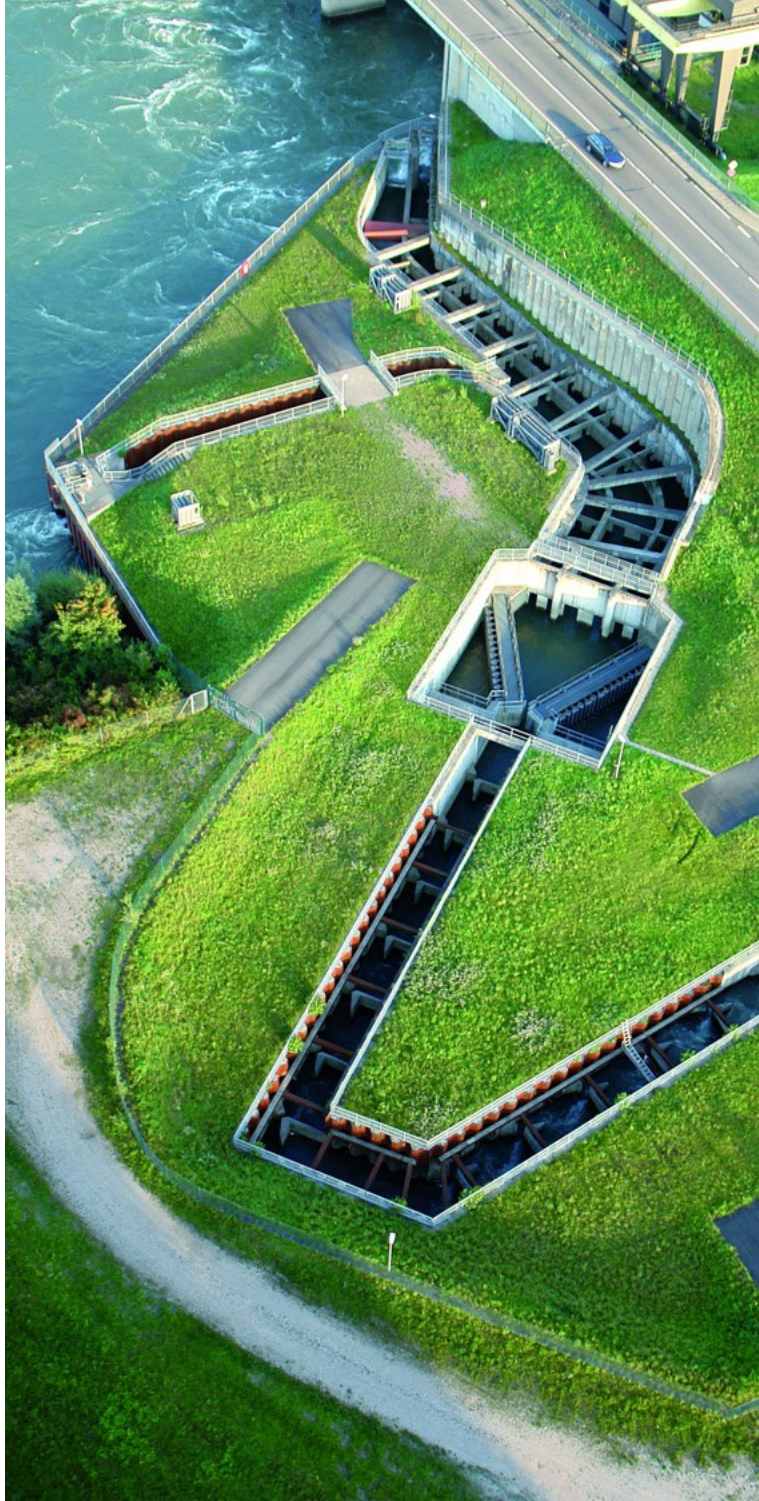
$$\widehat{p}_{i,t} = (1-\alpha)\widehat{v}_{i,t} + \alpha/N$$

- We predict $\widehat{y}_t = \sum_{i=1}^{N} \widehat{p}_{i,t} x_{i,t}$
- We observe y_t and update the pre-weight

$$\widehat{v}_{j,t+1} = \frac{\widehat{p}_{j,t}\, e^{-\eta\, \ell_{j,t}}}{\sum_{i=1}^{d} \widehat{p}_{i,t}\, e^{-\eta\, \ell_{i,t}}} \qquad \text{where} \quad \ell_{i,s} = 2(\widehat{y}_s - y_s)x_{i,s}$$

II. A good set of experts

June 2013 – STAT



Consider as heterogeneous experts as possible

Some ideas to get more variety inside the set of experts

- Consider heterogeneous prediction methods
- Create new experts from the same method thanks to boosting, bagging
- Vary the considered covariate: weather, calendar, . . .
- Specialize the experts: focus on specific situation (cloudy days,. . .) during the training

The dataset

The dataset includes 1 696 days from January 1, 2008 to June 15, 2012

- The electricity consumption of EDF customers
- Side information
  - weather: temperature, nebulosity, wind
  - temporal: date, EJP
  - loss of clients

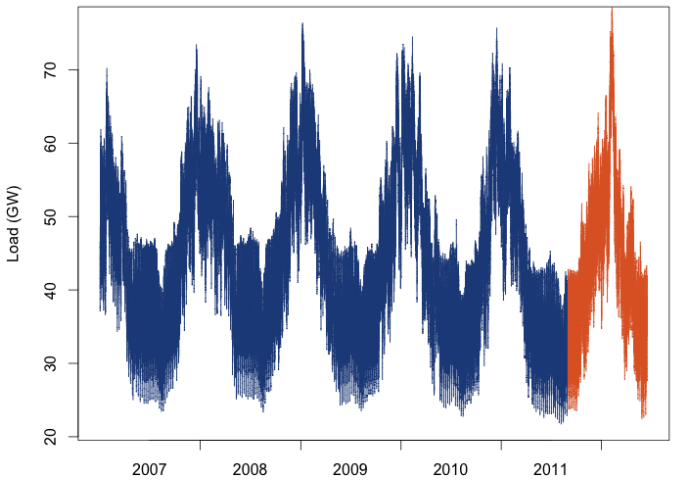
We remove uncommon days (public holidays ± 2) i.e., 55 days each year.

We split the dataset in two subsets

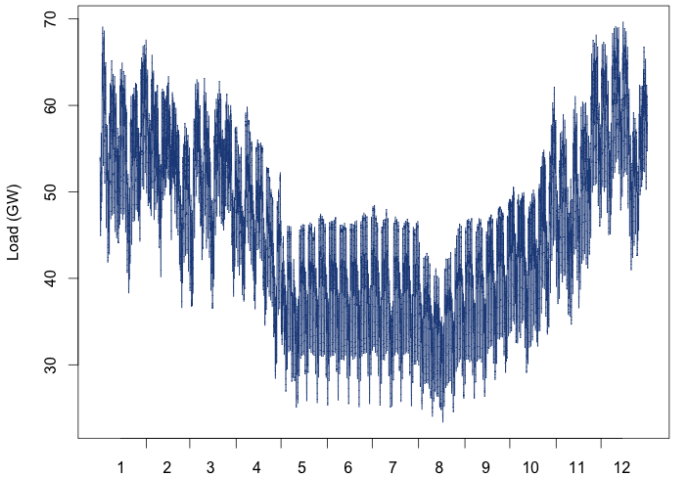
- Jan. 2008 – Aug. 2011: training set to build the experts
- Sept. 2011 – Jun. 2012: testing set

The dataset

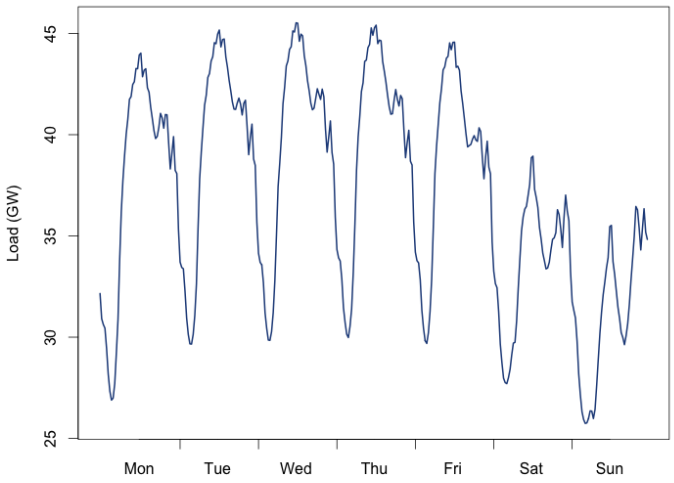
Load in training and testing sets



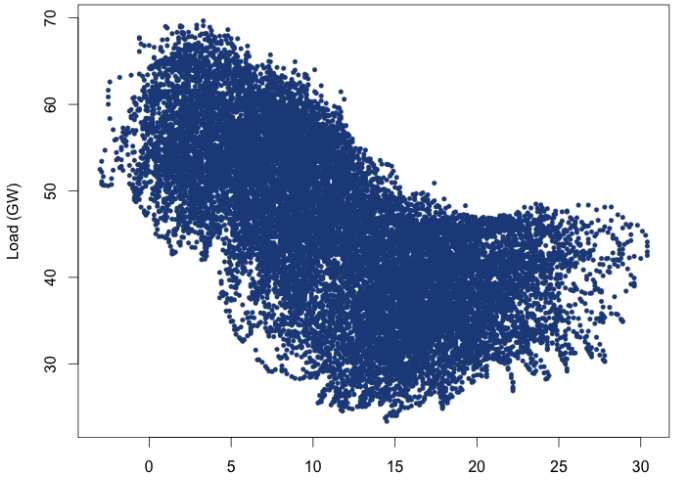
Load in 2008



Load during a week



Load according to the temperature (^{o}C)



Consider heterogeneous prediction methods

We kept 2 experts

- Gam – semi-parametric method

  Wood, 2006

- KWF – functional method based on similarity between days
  It does not use the temperature !

  Antoniadis, Brossat, Cugliari, and Poggi, COMPSTAT, 2010

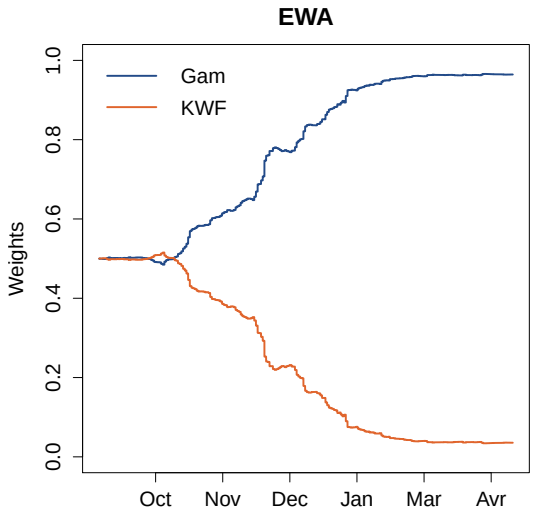
Other considered methods

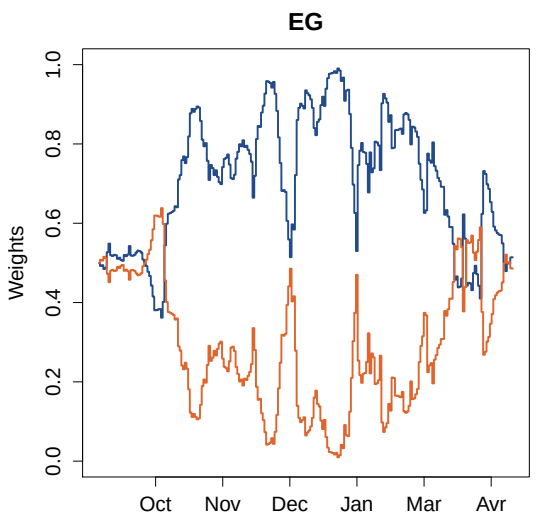
- Gam mid-term + short-term
- Boosting
- Random forests
- Regression trees

} Too similar with previous methods

Performance of the forecasting methods and of the aggregating algorithms

Method	RMSE (MW)
Best expert	847
Best convex vector	778
EWA	813
EG	778





Specializing the experts to diversify

Idea

Focus on specific scenarios during the training of the methods

Meteorological scenarios

- High / low temperature
- High / low variation of the temperature (since the last day, during the day)

Other scenarios

- High / low consumption
- Winter / summer

Such specialized experts suggest prediction only the days corresponding to their scenario

Specializing a method in cold days

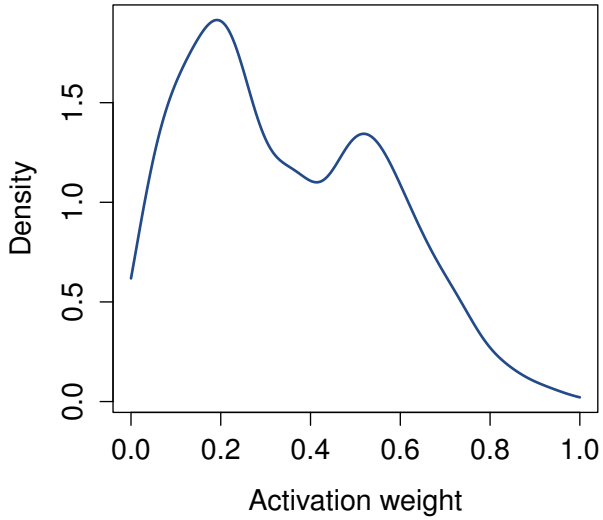
At day t, we consider

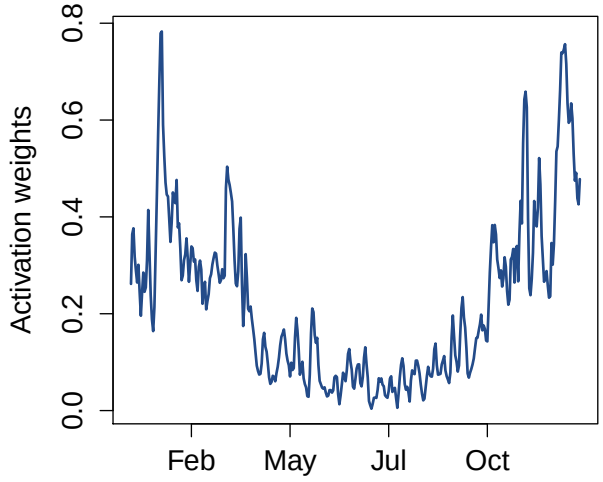
T_t = average temperature of the day

We normalize T_t on $[0, 1]$ and we choose for each day the weight

$$w_t = (1 - T_t)^2$$

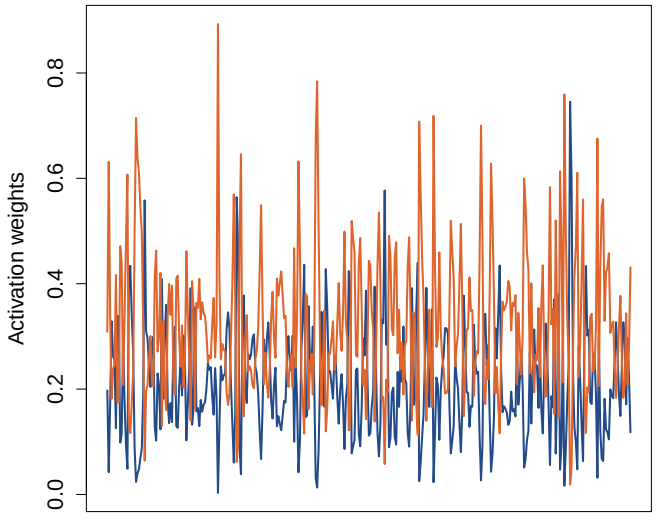
We then train our forecasting method using the prior weights w_t on the training days



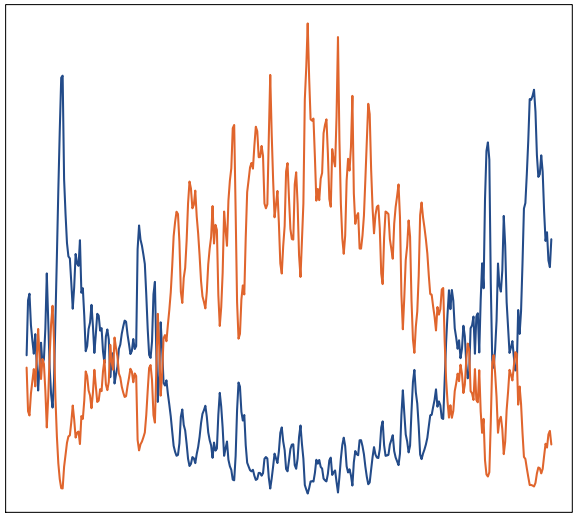


Weights given in 2008 for several specializing scenarios

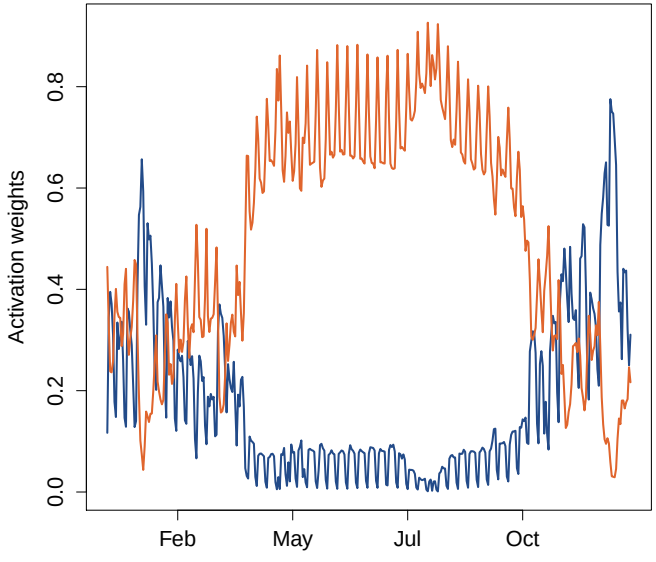
Difference of temp. with last day



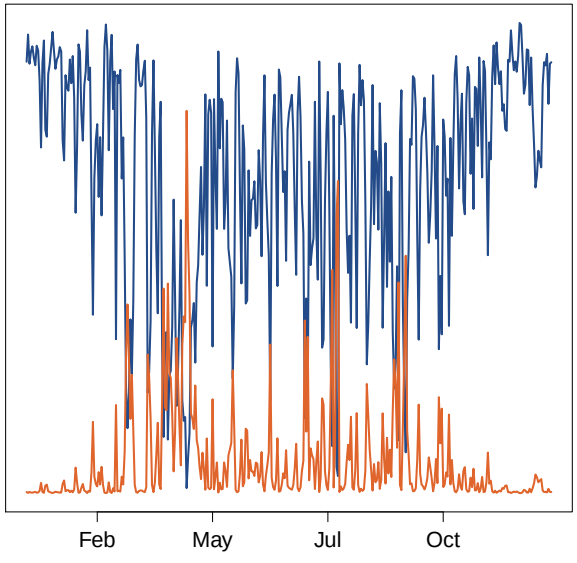
Hot / cold days



High / low consumption



Variation of temp. during the day



Aggregating experts that specialize

Setting

Each day some of the experts are active and output predictions (according to their specialization) while other experts do not

When the expert i is non active, we do not have access to its prediction

A solution is to assume that non active experts output the same prediction $\widehat{y}_t$ as we do and solve the fixed-point equation

$$\widehat{y}_t = \sum_{j \text{ active}} \widehat{p}_{j,t} x_{j,t} + \sum_{i \text{ non active}} \widehat{p}_{i,t} \widehat{y}_t$$

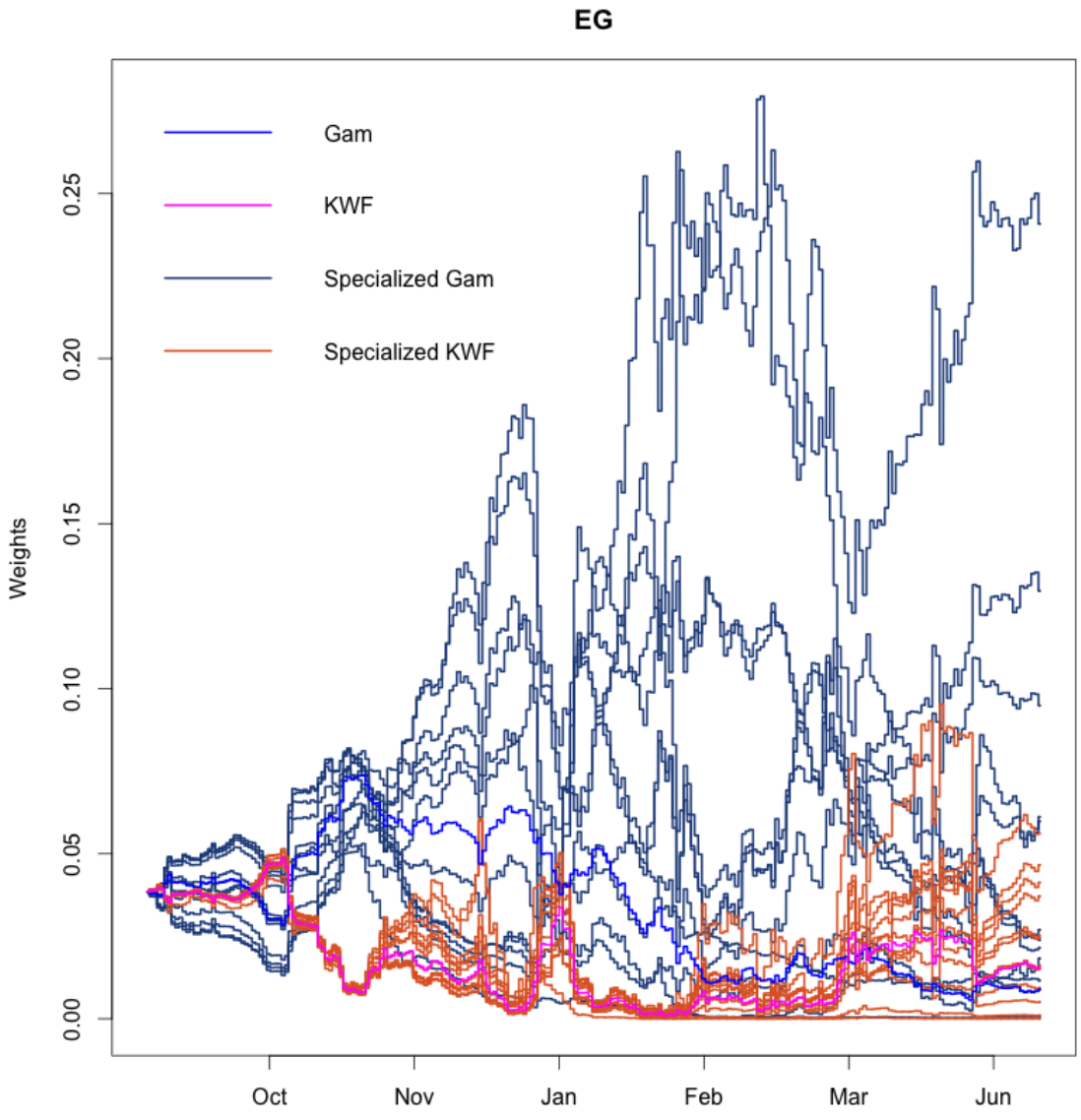
Can be extended to activation functions of the experts $\in [0, 1]$

Devaine, Gaillard, Goude, and Stoltz, Machine Learning, 2013

Performance of algorithms with specialized experts

2 base experts + 24 specialized experts

Method	RMSE (MW)
Best expert	804
Best convex vector	729
EWA (2 experts)	813
EG (2 experts)	778
EWA (26 experts)	765
EG (26 experts)	714



Performance of algorithms with specialized experts