Online hierarchical forecasting for power consumption data

Margaux Brégère, Malo Huard

Online Hierarchical Forecasting for Power Consumption Data

Margaux Brégère[a,b], Malo Huard[b]

[a] *EDF R&D, Palaiseau, France. INRIA - Département d'Informatique de l'École Normale Supérieure, PSL Research University, Paris, France.*
[b] *Laboratoire de mathématiques d'Orsay, CNRS, Université Paris-Saclay, Orsay, France.*

Abstract

This paper proposes a three-step approach to forecasting time series of electricity consumption at different levels of household aggregation. These series are linked by hierarchical constraints - global consumption is the sum of regional consumption, for example. First, benchmark forecasts are generated for all series using generalized additive models; second, for each series, the aggregation algorithm 'polynomially weighted average forecaster with multiple learning rates', introduced by Gaillard, Stoltz and van Erven in 2014, finds an optimal linear combination of the benchmarks; finally, the forecasts are projected onto a coherent subspace to ensure that the final forecasts satisfy the hierarchical constraints. By minimizing a regret criterion, we show that the aggregation and projection steps improve the root mean square error of the forecasts. Our approach is tested on household electricity consumption data; experimental results suggest that successive aggregation and projection steps improve the benchmark forecasts at different levels of household aggregation.
Keywords: Adjusting forecasts, Combining forecasts, Demand forecasting, Electricity, Time series

1. Introduction

Motivation: Electricity Forecasting. New opportunities come with the recent deployment of smart grids and the installation of meters: they record consumption quasi instantaneously in households. From these records, time series of demand are obtained at various levels of aggregation, such as consumption profiles and regions. For privacy reasons, household records may not be used directly. Moreover, consumption at individual level is erratic and difficult to predict. This is why we focus on household aggregations. For demand management, it is useful to predict the global consumption. Furthermore, to dispatch correctly the electricity into the grid, forecasting demand at a regional level is also an important goal. Finally, a good estimation of the consumption of some groups of

consumers (with the same profile) could be helpful for the electricity provider which may adapt its offer to perform effective demand side management. Thus, forecasts at various aggregated levels (entire population, geographical areas, groups of same consumption profiles) are useful for an efficient management of consumption. In this work, we first build at each aggregation level, and independently, benchmark forecasts using generalized additive models. Noticing that these time series may be correlated (the consumption of a given region may be close to the one of a neighboring region) and connected to each other through summation constraints (the global consumption is the sum of the region consumptions, e.g.), the problem considered falls under the umbrella of hierarchical time series forecasting (see, among others Hyndman, Ahmed, Athanasopoulos & Shang, 2011). Using these hierarchical relationships may improve the benchmark forecasts that were generated. Our approach consists in combining two methods: benchmark aggregation and projection in a constrained space. Our aim is to improve forecasts both at the global and at the local levels.

Literature Discussion for Hierarchical Forecasting. Traditionally two types of methods have been used for hierarchical forecasting: bottom-up and top-down approaches. In the bottom-up approaches (see Dunn, Williams & DeChaine, 1976) forecasts are constructed for lower-level quantities and are then summed up to obtain forecasts at the upper levels. In contrast, top-down approaches (see Gross & Sohl, 1990) work by forecasting aggregated quantities and then by determining disaggregate proportions to compute lower level predictions. Shlifer & Wolff (1979) compare these two families of methods and conclude that bottom-up approaches work better. Recently, it has indeed proven successful for load forecasting to improve the global consumption prediction error (see among others Auder, Cugliari, Goude & Poggi, 2018). Other approaches (neither bottom-up nor top-down) were recently introduced, for example Wickramasuriya, Athanasopoulos & Hyndman (2019) forecast all nodes in the hierarchy and reconcile (i.e. impose the respect of hierarchical constraints) them by projection. Their general MinT (for Minimum Trace) approach attempts to capture some cross-sectional information between times series via the covariance matrix of the errors of the base forecasts. It includes both oblique and orthogonal projections (this is discussed from a geometric perspective in Panagiotelis, Athanasopoulos, Gamakumara & Hyndman, 2020). Moreover, Van Erven & Cugliari (2015) introduce a game-theoretically optimal reconciliation method to improve a given set of forecasts. Firstly, one comes up with some forecasts for the time series without worrying about hierarchical constraints and then a reconciliation procedure is used to

make the forecasts aggregate consistent. This generalizes the previous orthogonal projection to other possible projections in the constrained space (which ensures that the forecasts satisfy the hierarchy). Most work on hierarchical forecasting concentrates on the mean, but some recent work has addressed probabilistic forecasting including that of Ben Taieb, Taylor & Hyndman (2017), Ben Taieb, Taylor & Hyndman (2020) and Panagiotelis et al. (2020).

Literature Discussion for Aggregation Methods. Aggregation methods (also called ensemble methods) for individual sequences forecasting originate from theoretical works by Vovk (1990), Cover (1991) and Littlestone & Warmuth (1994); their distinguishing feature with respect to classical ensemble methods is that they do not rely on any stochastic modeling of the observations and thus, are able to combine forecasts independently of their generating process. They have been proved to be very effective to predict time series (see for instance Mallet, Stoltz & Mauricette, 2009 and Devaine, Gaillard, Goude & Stoltz, 2013) and those methods were used to win forecasting competitions (see Gaillard, Goude & Nedellec, 2016). This aggregation approach has recently been extended to the hierarchical setting by Goehry, Goude, Massart & Poggi (2020); they used a bottom-up forecasting approach which consists in aggregating the consumption forecasts of small customers clusters.

In this article we combine the reconciliation approach based on orthogonal projection with some aggregation algorithm to propose a three stage meta-algorithm which is as follows:

1. Generate base forecasts for all times series in the hierarchy,

2. Apply, for each series, the aggregation algorithm ' that finds an optimal linear combination of the base forecasts

3. Project the combination forecasts onto a coherent subspace to ensure the final forecasts satisfy the hierarchical constraints.

The second step here provides the innovation (Steps 1 and 3 on their own are equivalent to the Ordinary Least Squares version of the MinT algorithm – see Wickramasuriya et al., 2019). By including an aggregation algorithm between these steps, much more of the cross-sectional information is able to be captured, thus improving the forecasts. A theoretical result is provided for the regret bound of the meta-algorithm which ensures that aggregation and projection steps improve the root mean square error of the forecasts. We then illustrate the proposed methods using smart meter data collected in Great Britain by multiple energy providers (see Schellong, 2011 and AECOM,

2018). '*Energy Demand Research Project*' data gathers multiple households power consumption data. We consider two population segmentations: a spatial segmentation based on the location of the households and a behavioral one based on household consumption profiles. For all aggregation levels, we generate benchmark forecasts using generalized additive models (see Wood, 2006) and use the polynomially weighted average forecaster with multiple learning rates (ML-Pol, see Gaillard, Stoltz & van Erven, 2014 and Gaillard, 2015) aggregation algorithm to combine these predictions. We evaluate the performance of four types of predictions: benchmarks, aggregated benchmarks, projected benchmarks and finally aggregated and projected benchmarks. Results show that the propose approach improve the root mean square error of the forecasts at the different levels of household aggregation.

Notation. Without further indications, $\|\mathbf{x}\|$ denotes the Euclidean norm of a vector $\mathbf{x}$. For the other norms, there will be a subscript: e.g., the $L1$-norm and the infinity norm of $\mathbf{x}$ are denoted by $\|\mathbf{x}\|_1$ and $\|\mathbf{x}\|_\infty$, respectively. Moreover, vectors will be in bold type and unless stated otherwise, they are column vectors, while matrices will be in bold underlined. We denote the inner product of two vectors $\mathbf{x}$ and $\mathbf{y}$ of the same size by $\mathbf{x} \cdot \mathbf{y} = \mathbf{x}^{\mathrm{T}}\mathbf{y}$. Finally, the cardinality of a finite set $\mathcal{D}$ is denoted by $|\mathcal{D}|$.

2. Methodology

With Γ, a set of aggregation levels (entire population, regions, behavioral clusters of households, e.g.), we consider the set of time series $\left\{(y_t^\gamma)_{t>0}, \gamma \in \Gamma\right\}$ connected to each other by some summation constraints: a few of them are equal to the sum of several others. To forecast these time series, a set of benchmark forecasts is generated. At any time step t, we want to forecast the vector of the values of the $|\Gamma|$ times series at t, denoted by $\mathbf{y}_t \stackrel{\text{def}}{=} (y_t^\gamma)_{\gamma\in\Gamma}$. For each node γ and each time step t, we will provide point forecasts $\widetilde{\mathbf{y}}_t^\gamma$ of y_t^γ and will focus on squared error losses $(y_t^\gamma - \widetilde{\mathbf{y}}_t^\gamma)$. We propose a method to obtain relevant forecasts from these benchmark forecasts.

2.1. Modeling of the Hierarchical Relationships

The relationships between the time series induce a hierarchy which should be exploited to improve forecasts. These summation constraints may be represented by one or more trees, the value at each node being equal to the sum of the ones at its leaves. Γ denotes the set of aggregation

levels which is also the set of the tree's nodes. There are as many summation constraints as there are nodes with leaves. Subsequently, we will introduce a matrix $\underline{\mathbf{K}}$ to encode these relationships. Each line of $\underline{\mathbf{K}}$ is related to one of the summation constraints with -1 at the associated node and 1 at its leaves. Thus, for any time step t, the vector of the values of the $|\Gamma|$ times series at t, denoted by $\mathbf{y}_t$, is in the kernel of $\underline{\mathbf{K}}$ (i.e. in the set of vectors $\mathbf{y}$ such as $\mathbf{y}\underline{\mathbf{K}} = \mathbf{0}_n$, where $\mathbf{0}_n$ is a vector of n zeros and n is the number of hierarchical constraints). Example 1 below treats a single summation constraint and Example 2 presents more complex relationships between the time series by considering two different partitions of the same time series. In our experiments, the underlying hierarchies will be of these two forms.

Example 1: Two-level Hierarchy. The simplest approach consists in considering a single equation connecting the time series. Here, y^{Tot} stands for the one which is the sum of the N others which are denoted by $y^1, \ldots, y^N$ (for example, the power consumption of a population of households which are distributed in N regions) The underlying hierarchy is represented in Figure 1 by a tree with a single root directly connected to N leaves. For any time step t, the time series satisfy $y_t^{\text{Tot}} = y_t^1 + y_t^2 + \cdots + y_t^N$ and the vector $\mathbf{y}_t = \left(y_t^{\text{Tot}}, y_t^1, \ldots, y_t^N\right)^{\text{T}}$ respects the hierarchy if and only if $\underline{\mathbf{K}}\mathbf{y}_t = 0$ with $\underline{\mathbf{K}} = \big(-1, 1, 1, \ldots, 1\big)$.

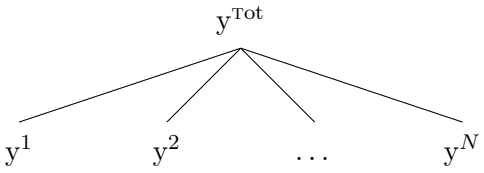


Figure 1: Representation of a two-level hierarchy.

Example 2: Two Crossed Hierarchies. Considering two partitions, the time series can

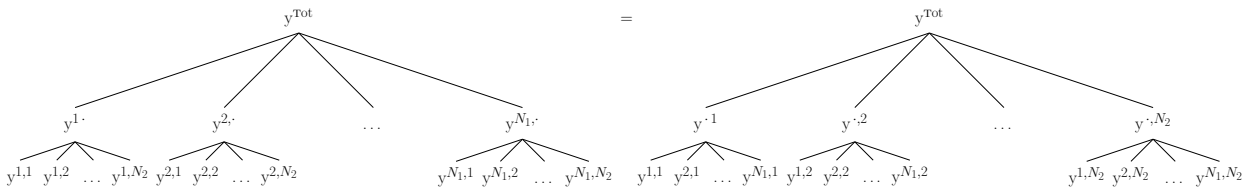


Figure 2: Representation of two crossed hierarchies.

be represented with two three-level trees sharing the same root and leaves. Only the intermediate levels differs according to which partition is firstly taking into account. The leaves of the trees form a $N_1 \times N_2$-matrix $\left(y^{i,j}\right)_{1\leqslant i\leqslant N_1,\, 1\leqslant j\leqslant N_2}$. An intermediate node of the first tree $y^{i,\cdot}$ is the sum of the line i while a node $y^{\cdot j}$ of the second tree is the sum of the column j. Whether we sum rows or columns first, the sum of all coefficients is y_t^{Tot}. One partition may refer to a geographic distribution of the households while the other classifies them according to their consumption behaviors. The first tree breaks down consumption firstly by the N_1 regions and then by the N_2 household profiles. The second one divides the households according to their habits before splitting them geographically. Both trees are represented in Figure 2. For any time step t, the time series satisfy the $2 + N_1 + N_2$ equations

$$
\begin{aligned}
y_t^{\text{Tot}} &= y_t^{1,\cdot} + y_t^{2,\cdot} + \cdots + y_t^{N_1,\cdot} && (1)\\
y_t^{i,\cdot} &= y_t^{i,1} + y_t^{i,2} + \cdots + y_t^{i,N_2}, \quad \forall\, i = 1, \ldots, N_1 && (2\text{i})\\
y_t^{\text{Tot}} &= y_t^{\cdot,1} + y_t^{\cdot,2} + \cdots + y_t^{\cdot,N_2} && (3)\\
y_t^{\cdot,j} &= y_t^{1,j} + y_t^{2,j} + \cdots + y_t^{N_1,j}, \quad \forall\, j = 1, \ldots, N_2. && (4\text{j})
\end{aligned}
$$

Equations (1) and (3) refer to the first level of the trees while the $N_1 + N_2$ Equations (2i) and (4j) refer to second levels. At a time step t, by ordering the time series in the vector as

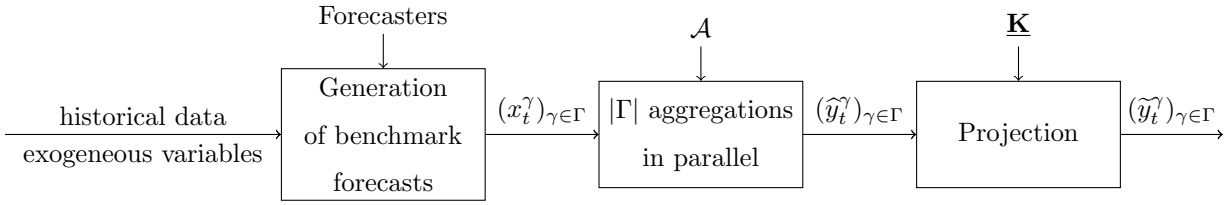
$$
\mathbf{y}_t = (y_t^{\text{Tot}}, y_t^{1,\cdot}, \ldots, y_t^{N_1,\cdot}, y_t^{\cdot,1}, \ldots, y_t^{\cdot,N_1}, y_t^{1,1}, y_t^{1,2}, \ldots, y_t^{N_1,N_2}),
$$

it may be seen that they respect the hierarchy if and only if $\underline{\mathbf{K}}\mathbf{y}_t = \mathbf{0}_{2+N_1+N_2}$ with

$$
\underline{\mathbf{K}} = \begin{pmatrix}
-1 & \overbrace{1 \;\cdots\; 1}^{N_1} & & & \overbrace{1 \;\cdots\; 1}^{N_2} & & & \leftarrow\ (1)\\
& -1 & & & & \ddots & & \leftarrow\ (2\text{i})\\
& & \ddots & & & & \overbrace{1 \;\cdots\; 1}^{N_2} & \leftarrow\ (2\text{i})\\
& & & -1 & & & & \leftarrow\ (2\text{i})\\
-1 & & & \overbrace{1 \;\cdots\; 1}^{N_2} & & & & \leftarrow\ (3)\\
& & & -1 & 1 & & 1 & \leftarrow\ (3)\\
& & & & \ddots & \ddots \quad \cdots \quad \ddots & & \leftarrow\ (4\text{j})\\
& & & & & -1 \qquad\quad 1 \qquad\quad 1 & &
\end{pmatrix}.
$$

2.2. A Three-step Forecast

For each node γ and time step t, we first, thanks to an historical data set of the time series and to some exogenous variables proper to the node γ, a forecaster makes the prediction x_t^γ. We propose to use the knowledge of all of these $|\Gamma|$ benchmark forecasts and of the summation constraints to improve the predictions. For each node γ and time step t, we then form our prediction $\widehat{y}_t^\gamma$ by linearly combining the components of the base forecasts $(x_t^i)_{i\in\Gamma}$ thanks to a so-called aggregation algorithm (a copy $\mathcal{A}^\gamma$ of an aggregation algorithm $\mathcal{A}$ is run separately for each node γ). That is, we use all $|\Gamma|$ benchmark forecasts to predict y_t^γ, not only x_t^γ. Finally, a reconciliation step will update the forecast vector so that it is in the kernel $\underline{\mathbf{K}}$. Let us denote by $\widetilde{y}_t^\gamma$ the final forecasts. We are now going to detail each step of the procedure schematized below.



First Step: Generation of benchmark forecasts. For each node $\gamma \in \Gamma$, at each time step t, thanks to an historical data set of the time series $(y_s^\gamma)_{1\leqslant s\leqslant t-1}$ and to some exogenous variables proper to γ, a forecaster makes the prediction x_t^γ. The forecasting method we use in the experiments is based on generalized additive models. These $|\Gamma|$ benchmark forecasts collected into a vector $\mathbf{x}_t = \left(x_t^\gamma\right)_{\gamma\in\Gamma}$. The benchmark vector $\mathbf{x}_t$ is used in the aggregation step that comes next to predict again each time series. We focus here on $|\Gamma|$ benchmark forecasts – one for each of the nodes; however, we could also have considered several predictions per nodes.

Second Step: Aggregation. The above benchmark forecasts are generated independently with different exogenous variables and possibly different methods. Yet, the observations $\left(y_t^\gamma\right)_{\gamma\in\Gamma}$ may be correlated. For example, considering load forecasting, the consumptions associated with two nearby regions can be strongly similar. Furthermore, the observations are related though the summation constraints (although we disregard these equations here). This is why linearly combining the benchmark forecasts may refine some forecasts – this is exactly what this step does. Formally, an aggregation algorithm outputs at each round a vector of weights $\widehat{\mathbf{u}}_t^\gamma$ and returns the forecast $\widehat{y}_t^\gamma \stackrel{\text{def}}{=} \widehat{\mathbf{u}}_t^\gamma \cdot \mathbf{x}_t$. It does so based on the information available, that is, a feature vector $\mathbf{x}_t$ made of benchmark forecasts in our case and past data. We consider an aggregation algorithm $\mathcal{A}$ and form

a copy $\mathcal{A}^\gamma$ for each node γ, which we feed with an input parameter vector $\mathbf{s}_0^\gamma$. These predictions are then gathered into the vector $\widehat{\mathbf{y}}_t = (y_t^\gamma)_{\gamma\in\Gamma}$.

Instead of this approach based on benchmark forecasting and aggregation node by node, we could have considered a meta-model to directly predict the time series vector $\left(y_t^\gamma\right)_{\gamma\in\Gamma}$ at each time step t (with a common forecaster and therefore without any aggregation step). Once this global forecast would have been obtained, we would have gone straight to the projection stage. In such a model, the number of variables to be taken into account (the historical data of the time series but also the exogenous variables specific to each node) would have been considerable and getting relevant forecasts would have not been an easy task. But actually, a practical choice motivated our method for the most. Indeed, the forecasters may be black boxes proper to each node and the exogenous variables of a node γ may be unknown at a node γ'. In our experiments, we followed this three-step approach; however, our method totally operates if, for each node γ and at each time step t, an external expert provides the forecast x_t^γ. How these benchmark forecasts have been obtained is no longer an issue and the aim is to improve these benchmark forecasts with aggregation and projection steps. Thus, at each time step t, only the benchmark forecasts are revealed at time t and by skipping the generation of benchmark step, we go straight to the aggregation step.

Third Step: Projection. As the $|\Gamma|$ executions of Algorithm $\mathcal{A}$ are run in parallel and independently, the obtained forecast vector $\widehat{\mathbf{y}}_t$ does not necessarily respect hierarchical constraints. To correct that, we consider the orthogonal projection of $\widehat{\mathbf{y}}_t$ onto the kernel of $\underline{\mathbf{K}}$, which we denote by $\Pi_{\underline{\mathbf{K}}}(\widehat{\mathbf{y}}_t)$. As the orthogonal projection matrix onto $\underline{\mathbf{K}}$ is given by $\underline{\mathbf{K}}^{\mathrm{T}}(\underline{\mathbf{K}}\underline{\mathbf{K}}^{\mathrm{T}})^{-1}\underline{\mathbf{K}}$, the projection onto the kernel of $\underline{\mathbf{K}}$ is $\Pi_{\underline{\mathbf{K}}} = \left(I_{|\Gamma|} - \underline{\mathbf{K}}^{\mathrm{T}}(\underline{\mathbf{K}}\underline{\mathbf{K}}^{\mathrm{T}})^{-1}\underline{\mathbf{K}}\right)$. The updated forecast $\widetilde{\mathbf{y}}_t \overset{\text{def}}{=} \Pi_{\underline{\mathbf{K}}}(\widehat{\mathbf{y}}_t)$ fulfills the hierarchical constraints.

To sum up, at each time step t, we first generate benchmark forecasts $\mathbf{x}_t$. These predictions are then aggregated to form a new vector of forecast $\widehat{\mathbf{y}}_t$, which is itself updated in the projection step in $\widetilde{\mathbf{y}}_t$. This procedure is stated in Meta-algorithm 1. Moreover, we can also directly project the benchmark forecasts, skipping the aggregation step; this leads to the forecasts $\Pi_{\underline{\mathbf{K}}}(\mathbf{x}_t)$ – they are identical to the OLS version of the MinT algorithm proposed in Wickramasuriya et al. (2019). Thus, we get four forecasts ($\mathbf{x}_t$, $\Pi_{\underline{\mathbf{K}}}(\mathbf{x}_t)$, $\widehat{\mathbf{y}}_t$ and $\widetilde{\mathbf{y}}_t$) for each node and each time step. The performance of our strategies is measured in mean squared error. In the experiments (Section 5), we compare these four methods in the scope of power consumption forecasting.

Meta-algorithm 1 Generation, aggregation and projection for hierarchical forecasting

Input

Set of nodes Γ and constraint matrix $\underline{\mathbf{K}}$

Benchmark forecast generation method

Aggregation algorithm $\mathcal{A}$ taking parameter vector $\mathbf{s}_0$

Compute the orthogonal projection matrix $\Pi_{\underline{\mathbf{K}}} = \left(I_{|\Gamma|} - \underline{\mathbf{K}}^{\mathrm{T}}(\underline{\mathbf{K}}\underline{\mathbf{K}}^{\mathrm{T}})^{-1}\underline{\mathbf{K}}\right)$

for $\gamma \in \Gamma$ **do**

Create a copy of $\mathcal{A}$ denoted by $\mathcal{A}^\gamma$ and run with $\mathbf{s}_0^\gamma$

for $t = 1, \ldots, T$ **do**

Generate benchmark forecasts $\mathbf{x}_t$

for $\gamma \in \Gamma$ **do**

$\mathcal{A}^\gamma$ outputs $\widehat{y}_t^\gamma = \mathbf{u}_t^\gamma \cdot \mathbf{x}_t$

Collect forecasts: $\widehat{\mathbf{y}}_t = (\widehat{y}_t^\gamma)_{\gamma\in\Gamma}^{\mathrm{T}}$

Project forecasts: $\widetilde{\mathbf{y}}_t = \Pi_{\underline{\mathbf{K}}}(\widehat{\mathbf{y}}_t)$

for $\gamma \in \Gamma$ **do**

$\mathcal{A}^\gamma$ observes y_t^γ

Suffer a prediction error $\frac{1}{|\Gamma|}\sum_{\gamma\in\Gamma}\left(y_t^\gamma - \widetilde{y}_t^\gamma\right)^2$

aim

Minimize the average prediction error $\widetilde{L}_T = \dfrac{1}{T}\displaystyle\sum_{t=1}^{T}\frac{1}{|\Gamma|}\left\|\mathbf{y}_t - \widetilde{\mathbf{y}}_t\right\|^2 = \frac{1}{T\,|\Gamma|}\sum_{t=1}^{T}\sum_{\gamma\in\Gamma}\left(y_t^\gamma - \widetilde{y}_t^\gamma\right)^2.$

2.3. Assessment of the Forecasts – Form of the Theoretical Guaranties Achieved

Our final forecasts are linear combinations of the benchmark forecasts and, after $T \geqslant 1$ time steps, they are evaluated by the average prediction error

$$\widetilde{L}_T \stackrel{\text{def}}{=} \frac{1}{T}\sum_{t=1}^{T}\frac{1}{|\Gamma|}\sum_{\gamma\in\Gamma}\left(y_t^\gamma - \widetilde{y}_t^\gamma\right)^2. \tag{1}$$

We want to compare our method to constant linear combinations of benchmark forecasts; namely, to the strategies defined by constant combination vectors $(\mathbf{u}^\gamma)_{\gamma\in\Gamma}$ which provide, for any node γ and any time step t, the forecasts $\mathbf{u}^\gamma \cdot \mathbf{x}_t$. For example, recalling that, for $\gamma \in \Gamma$, x_t^γ is the benchmark prediction of y_t^γ, using the standard basis vector that points in the γ direction $\boldsymbol{\delta}^\gamma \stackrel{\text{def}}{=} \mathbf{1}_{\{i=\gamma\}}$ as weights should be a good first choice to define a constant linear combination. Indeed, this strategy

provides, for any $\gamma \in \Gamma$, $\boldsymbol{\delta}^\gamma \cdot \mathbf{x}_t = x_t^\gamma$ as forecast for y_t^γ. Thus, the matrix $\left(\boldsymbol{\delta}^\gamma\right)_{\gamma\in\Gamma}$ defines a constant benchmark strategy and its cumulative prediction error, after $T \geqslant 1$ time steps, is

$$L_T\Big(\left(\boldsymbol{\delta}^\gamma\right)_{\gamma\in\Gamma}\Big) \stackrel{\text{def}}{=} \frac{1}{T}\sum_{t=1}^{T}\frac{1}{|\Gamma|}\sum_{\gamma\in\Gamma}\left(y_t^\gamma - x_t^\gamma\right)^2. \tag{2}$$

As soon as the benchmark forecasts $(x_t^\gamma)_{\gamma\in\Gamma}$ are well-chosen, this quantity is small. But, these benchmark predictions do not satisfy the summation constraints *a priori* and it will not be fair to compare our forecasts (which do respect to hierarchy – projection step ensures it) to these benchmark forecasts – or any other constant linear combinations of benchmarks. Thus, we are going to introduce the set $\mathcal{C}$ which contains all the constant strategies which satisfy the hierarchical constraints and detail how such a strategy can be represented by a $|\Gamma| \times |\Gamma|$-matrix $\underline{\mathbf{U}} \in \mathcal{C}$. Then, we will decompose, for any $\underline{\mathbf{U}} \in \mathcal{C}$, the average prediction error into an approximation error $L_T(\underline{\mathbf{U}})$ and a sequential estimation error $\mathcal{E}_T(\underline{\mathbf{U}})$. To ensure that our strategy does almost as well as the best constant combination of benchmark forecasts, we want to get a guarantee on the average prediction error $\widetilde{L}_T$, defined in Equation (1), of form:

$$\widetilde{L}_T \leqslant \inf_{\underline{\mathbf{U}}\in\mathcal{C}}\Big\{L_T(\underline{\mathbf{U}}) + \mathcal{E}_T(\underline{\mathbf{U}})\Big\}, \quad \text{where} \quad \mathcal{E}_T(\underline{\mathbf{U}}) = \mathcal{O}\Big(\frac{1}{\sqrt{T}}\Big). \tag{3}$$

Indeed, if $\mathcal{E}_T(\underline{\mathbf{U}}) \underset{T\to+\infty}{\longrightarrow} 0$, the average prediction error of our strategy tends to $L_T(\underline{\mathbf{U}})$ and classical convergence rate are in $\frac{1}{\sqrt{T}}$. We will explain how this aim is equivalent to minimizing a quantity called regret.Such a result ensures that our strategy outperforms all $\underline{\mathbf{U}} \in \mathcal{C}$ strategies, therefore the set $\mathcal{C}$ is called "class of comparison". In this work, it refers to the set of constant strategies that satisfy hierarchical constraints and is defined just below.

2.3.1. Class of Comparison

In what follows, the kernel and the image of a matrix $\underline{\mathbf{A}} \in \mathbb{R}^{n\times m}$ are denoted by $\mathrm{Ker}(\underline{\mathbf{A}}) \stackrel{\text{def}}{=} \left\{\mathbf{u} \in \mathbb{R}^m \,|\, \underline{\mathbf{A}}\mathbf{u} = \mathbf{0}_n\right\}$ (with $\mathbf{0}_n$ a vector of n zeros) and $\mathrm{Im}(\underline{\mathbf{A}}) \stackrel{\text{def}}{=} \left\{\underline{\mathbf{A}}\mathbf{u} \,|\, \mathbf{u} \in \mathbb{R}^m\right\}$, respectively. We consider here a constant strategy, namely $|\Gamma|$ linear combinations of the benchmark forecasts. More formally, let us denote by $\mathbf{u}^\gamma$ a constant weight vector which provides, for any time step t, the forecast $\mathbf{u}^\gamma \cdot \mathbf{x}_t$ for the time series y_t^γ. By batching these $|\Gamma|$ vectors into $\underline{\mathbf{U}} \stackrel{\text{def}}{=} (\mathbf{u}^\gamma)_{\gamma\in\Gamma} \in \mathcal{M}_{|\Gamma|}$, predictions satisfy the constraints for a time step t if $\underline{\mathbf{U}}^{\mathrm{T}}\mathbf{x}_t \in \mathrm{Ker}(\underline{\mathbf{K}})$. For it to be true for any t (except for a few special cases, such as if all benchmark vectors are null), this requires that the

image of $\underline{\mathbf{U}}^{\mathrm{T}}$ is in the kernel of $\underline{\mathbf{K}}$. We introduce the following set of matrices, for which associated forecasts necessarily satisfy the hierarchical constraints

$$\mathcal{C} \stackrel{\text{def}}{=} \left\{ \underline{\mathbf{U}} = \left(\mathbf{u}^1 \,\middle|\, \ldots \,\middle|\, \mathbf{u}^{|\Gamma|}\right) \middle|\, \mathrm{Im}\left(\underline{\mathbf{U}}^{\mathrm{T}}\right) \subset \mathrm{Ker}\left(\underline{\mathbf{K}}\right) \right\}. \tag{4}$$

Note that, for any matrix $\underline{\mathbf{U}} \in \mathcal{M}_{|\Gamma|}$, by definition of the orthogonal projection $\Pi_{\underline{\mathbf{K}}}$, the forecast vector $\Pi_{\underline{\mathbf{K}}}\underline{\mathbf{U}}^{\mathrm{T}}\mathbf{x}_t$ satisfies the hierarchical relationships so the set $\mathcal{C}$ contains the matrix $\underline{\mathbf{U}}\Pi_{\underline{\mathbf{K}}}^{\mathrm{T}}$. This implies that the set $\mathcal{C}$ is not empty. To compare our methods to any constant strategy $\underline{\mathbf{U}} \in \mathcal{C}$, we now introduce the common notion of regret.

2.3.2. Aim: Regret Minimization

We want to compare the average prediction error $\widetilde{L}_T$ to $L_T(\underline{\mathbf{U}})$, where $\underline{\mathbf{U}} \in \mathcal{C}$ so the forecasts associated with $\underline{\mathbf{U}}$ satisfy the hierarchical constraints – otherwise, the two strategies would not be comparable because our predictions do respect the hierarchy. Good algorithms should ensure that $\widetilde{L}_T$ is not too far from the best $L_T(\underline{\mathbf{U}})$. We thus define, for any $\underline{\mathbf{U}} = (\mathbf{u}^\gamma)_{\gamma\in\Gamma} \in \mathcal{C}$, the cumulative prediction error of the associated constant linear combinations of benchmarks by

$$L_T(\underline{\mathbf{U}}) \stackrel{\text{def}}{=} \frac{1}{T}\sum_{t=1}^{T}\frac{1}{|\Gamma|}\sum_{\gamma\in\Gamma}\left(y_t^\gamma - \mathbf{u}^\gamma\cdot\mathbf{x}_t\right)^2 = \frac{1}{T|\Gamma|}\sum_{t=1}^{T}\left\|\mathbf{y}_t - \underline{\mathbf{U}}^{\mathrm{T}}\mathbf{x}_t\right\|^2. \tag{5}$$

In order to obtain a theoretical guarantee of the form of Equation (3), we decompose the average prediction error as

$$\widetilde{L}_T = L_T(\underline{\mathbf{U}}) + \frac{R_T(\underline{\mathbf{U}})}{T|\Gamma|}, \tag{6}$$

so, the aim for algorithms to get $\widetilde{L}_T$ as close as possible to $\min_{\underline{\mathbf{U}}\in\mathcal{C}} L_T(\underline{\mathbf{U}})$ (with $\mathcal{C}$ the class of comparison defined above), is equivalent to

$$\max_{\underline{\mathbf{U}}\in\mathcal{C}} R_T(\underline{\mathbf{U}}) = T|\Gamma| \times \left(\widetilde{L}_T - \min_{\underline{\mathbf{U}}\in\mathcal{C}} L_T(\underline{\mathbf{U}})\right) \tag{7}$$

being small. The quantity $R_T(\underline{\mathbf{U}})$, commonly called regret is defined as the difference between the cumulative prediction error of our method and the one for weights $\underline{\mathbf{U}}$:

$$R_T(\underline{\mathbf{U}}) \stackrel{\text{def}}{=} T|\Gamma| \times \left(\widetilde{L}_T - L_T(\underline{\mathbf{U}})\right) = \sum_{t=1}^{T}\left\|\mathbf{y}_t - \widetilde{\mathbf{y}}_t\right\|^2 - \sum_{t=1}^{T}\left\|\mathbf{y}_t - \underline{\mathbf{U}}^{\mathrm{T}}\mathbf{x}_t\right\|^2. \tag{8}$$

In the light of Equation (6), the average prediction error $\widetilde{L}_T$ we attempt to minimize breaks down into an approximation error $L_T(\underline{\mathbf{U}})$ (the best prediction error we can hope for) and a sequential estimation error (dependent on how quickly the model estimates $\underline{\mathbf{U}}$), proportional to the regret $R_T(\underline{\mathbf{U}})$.

The introduction of a regret criteria is very common for online forecasting methods (see, among others, Devaine et al., 2013 and Mallet et al., 2009), and for an algorithm to be useful, $\max_{\underline{\mathbf{U}}\in\mathcal{C}} R_T(\underline{\mathbf{U}})$ need to be sub-linear in T (otherwise the error remains constant – or even worse: it increases with time). Typical theoretical guaranties provide bounds of order $\sqrt{T}$ (see for example, Deswarte, Gervais, Stoltz & Da Veiga, 2018 and Amat, Michalski & Stoltz, 2018).

3. Main Theoretical Result

From now on, let us introduce the following notation concerning the regret bound of Algorithm $\mathcal{A}$.

Assumption 1. *We assume that, for any set $\mathcal{D} \in \mathbb{R}^{|\Gamma|}$, for any $\gamma \in \Gamma$ with the initialization parameter vector $\mathbf{s}_0^\gamma$, for $T > 0$, any $\mathbf{x}_{1:T} = \mathbf{x}_1, \dots \mathbf{x}_T$ and any $y_{1:T}^\gamma = y_1^\gamma, \dots, y_T^\gamma$, Algorithm $\mathcal{A}^\gamma$ provides a regret bound of the following form:*

$$R_T^\gamma(\mathcal{D}) \stackrel{\text{def}}{=} \sum_{t=1}^{T} \left(y_t^\gamma - \widehat{y}_t^\gamma\right)^2 - \min_{\mathbf{u}^\gamma\in\mathcal{D}} \sum_{t=1}^{T} \left(y_t^\gamma - \mathbf{u}^\gamma \cdot \mathbf{x}_t\right)^2 \leqslant \mathrm{B}\big(\mathbf{x}_{1:T}^\gamma, y_{1:T}^\gamma, \mathbf{s}_0^\gamma\big). \tag{9}$$

As getting a linear bound is trivial (by using the common assumption that prediction errors are bounded), the bounds $B(\dots)$ have to be sub-linear to be of interest. We will see that the algorithm we used in the experiments ensures such a regret bound. This assumption makes it possible to establish a bound of the cumulative regret.

Theorem 1. *For any sets $\mathcal{D} \in \mathbb{R}^{|\Gamma|}$ and $\mathcal{B} \subset \mathcal{M}_{|\Gamma|}$, by denoting, $\mathcal{B}_{|\mathcal{D}} \stackrel{\text{def}}{=} \big\{\underline{\mathbf{U}} \in \mathcal{B} \,|\, \forall\gamma \in \Gamma,\, \mathbf{u}^\gamma \in \mathcal{D}\big\}$, under Assumption 1, for any $T \geqslant 1$, the cumulative regret satisfies*

$$R_T(\mathcal{D}) \stackrel{\text{def}}{=} \max_{\underline{\mathbf{U}}\in\mathcal{C}_{|\mathcal{D}}} R_T(\underline{\mathbf{U}}) = \sum_{t=1}^{T} \left\|\mathbf{y}_t - \widetilde{\mathbf{y}}_t\right\|^2 - \min_{\underline{\mathbf{U}}\in\mathcal{C}_{|\mathcal{D}}} \sum_{t=1}^{T} \left\|\mathbf{y}_t - \underline{\mathbf{U}}^{\mathrm{T}}\mathbf{x}\right\|^2 \leqslant \sum_{\gamma\in\Gamma} \mathrm{B}\big(\mathbf{x}_{1:T}^\gamma, y_{1:T}^\gamma, \mathbf{s}_0^\gamma\big). \tag{10}$$

The regret $R_T(\mathcal{D})$ is not just the sum over all the nodes of the regrets $R_T^\gamma(\mathcal{D})$ of Equation (9). Indeed, we do not evaluate the forecasts $\widehat{\mathbf{y}}_t$ here but those obtained after the projection step: $\widetilde{\mathbf{y}}_t$. The projection step provides a diminishing of the square prediction error and we just have to sum Equation (9) on all nodes to get the bound.

Proof. This regret bound results from two main arguments: Pythagorean theorem, on the one hand, and Assumption 1, on the other hand. For any $t \geqslant 1$, as $\mathbf{y}_t \in \mathrm{Ker}(\underline{\mathbf{K}})$, the Pythagorean theorem ensures

$$\left\|\mathbf{y}_t - \widetilde{\mathbf{y}}_t\right\|^2 = \left\|\mathbf{y}_t - \Pi_{\underline{\mathbf{K}}}(\widehat{\mathbf{y}}_t)\right\|^2 \leqslant \left\|\mathbf{y}_t - \widehat{\mathbf{y}}_t\right\|^2. \tag{11}$$

Let us fix a matrix $\underline{\mathbf{U}} = \left(\mathbf{u}^1 | \ldots | \mathbf{u}^{|\Gamma|}\right) \in \mathcal{C}_{|\mathcal{D}}$. Firstly, the application of Pythagorean theorem ensures that the projection step reduces regret. Rewriting the regret as a sum over the nodes, we then use Assumption 1 independently for each node of Γ to conclude the proof.

$$\begin{aligned}
\sum_{t=1}^{T} \left\|\mathbf{y}_t - \widetilde{\mathbf{y}}_t\right\|^2 - \sum_{t=1}^{T} \left\|\mathbf{y}_t - \underline{\mathbf{U}}^{\mathrm{T}}\mathbf{x}_t\right\|^2 &\overset{(11)}{\leqslant} \sum_{t=1}^{T} \left\|\mathbf{y}_t - \widehat{\mathbf{y}}_t\right\|^2 - \sum_{t=1}^{T} \left\|\mathbf{y}_t - \underline{\mathbf{U}}^{\mathrm{T}}\mathbf{x}_t\right\|^2 \\
&= \sum_{\gamma\in\Gamma}\sum_{t=1}^{T} \left(y_t^\gamma - \widehat{y}_t^\gamma\right)^2 - \sum_{\gamma\in\Gamma}\sum_{t=1}^{T} \left(y_t^\gamma - \mathbf{u}^\gamma \cdot \mathbf{x}_t\right)^2 \\
&\leqslant \sum_{\gamma\in\Gamma} \max_{\mathbf{u}^\gamma\in\mathcal{D}} \left(\sum_{t=1}^{T} \left(y_t^\gamma - \widehat{y}_t^\gamma\right)^2 - \sum_{t=1}^{T} \left(y_t^\gamma - \mathbf{u}^\gamma \cdot \mathbf{x}_t\right)^2 \right) = \sum_{\gamma\in\Gamma} R_T^\gamma(\mathcal{D}) \\
&\overset{(9)}{\leqslant} \sum_{\gamma\in\Gamma} \mathrm{B}\big(\mathbf{x}_{1:T}, y_{1:T}^\gamma, \mathbf{s}_0^\gamma\big).
\end{aligned} \tag{12}$$

As this bound is satisfied for all $\underline{\mathbf{U}} = \left(\mathbf{u}^1 | \ldots | \mathbf{u}^{|\Gamma|}\right) \in \mathcal{C}_{|\mathcal{D}}$, we conclude the proof by taking the minimum onto $\mathcal{C}_{|\mathcal{D}}$.

□

Note that similar results, also based on Pythagorean theorem, have been obtained in Panagiotelis et al. (2020) (see Theorem 3.1).

4. An example of an Aggregation Algorithm: Polynomially Weighted Average Forecaster with Multiple Learning rates (ML-Pol)

At a time step t, for a node $\gamma \in \Gamma$, a copy $\mathcal{A}^\gamma$ of an aggregation algorithm $\mathcal{A}$ takes the benchmark vector $\mathbf{x}_t$, which contains the predictions of all the nodes (including that of the considering node), as an input and outputs a weight vector $\mathbf{u}_t^\gamma$ and thus the forecast y_t^γ with $\mathbf{u}_t^\gamma \cdot \mathbf{x}_t$. We remind that the benchmark forecasts $\left(x_t^\gamma\right)_{\gamma\in\Gamma}$ are generated independently with possibly different exogenous variables but that the observations $\left(y_t^\gamma\right)_{\gamma\in\Gamma}$ may be strongly correlated. This is why we

consider aggregation to refine some forecasts by combining the benchmarks. Our experiments will demonstrate that this aggregation step improves the forecasts.

Benchmark forecasts and observation are firstly standardized according to a pre-processing fully described in Appendix A. The way in which this normalization modifies the theoretical results is also explained in the appendix. For the ease of notation, transformed benchmarks and observations will still be denoted by $\mathbf{x}_t$ and $\mathbf{y}_t$, respectively.

The algorithm comes with some theoretical guarantees of the form of Assumption 1 and we adapt Theorem 1 by replacing the bound $B(\dots)$ with the one provided by the theory. The latter was established under some boundedness assumptions on benchmark forecasts and observations, so we make the following assumption.

Assumption 2. *Boundedness assumptions. For any $t > 0$ and any $\gamma \in \Gamma$ we assume that there is a constant $C > 0$ such that*

$$|y_t^\gamma| \leqslant C \quad \text{and} \quad |x_t^\gamma| \leqslant C. \tag{13}$$

Here, the constant is common to all the nodes, which makes practically sense because of the pre-processing described in Appendix A.

The algorithm we consider was initially designed to compete against the best benchmark forecast. Namely, for a node $\gamma \in \Gamma$, polynomially weighted average forecaster with multiple learning rates (ML-Pol, see Gaillard et al., 2014 and Gaillard, 2015) provides some bound on the difference between the cumulative prediction error $L_T^\gamma \stackrel{\text{def}}{=} \sum_{t=1}^T \left(y_t^\gamma - \widehat{y}_t^\gamma\right)^2$ of the strategy and $\min_{i\in\Gamma} \sum_{t=1}^T \left(y_t^\gamma - x_t^i\right)^2$. At each time step t, the strategy computes weight vector $\mathbf{u}_t^\gamma = \left(u_t^{\gamma,i}\right)_{i\in\Gamma}$ based on historical data. These vectors are in the $|\Gamma|$-simplex, which we denote by $\Delta_{|\Gamma|}$. For each benchmark forecast $i \in \Gamma$, the weight $u_t^{\gamma,i}$ is a polynomial function of the cumulative prediction error of x_t^i. However, by using gradients of prediction errors instead of the original prediction errors, the average error of this algorithm may come close to $\min_{\mathbf{u}^\gamma\in\Delta_{|\Gamma|}} \frac{1}{T}\sum_{t=1}^T \left(y_t^\gamma - \mathbf{u}^\gamma \cdot \mathbf{x}_t\right)^2$. This "gradient trick" (see Cesa-Bianchi & Lugosi, 2006, Section 2.5) is already integrated in the statement of the algorithm provided in Appendix B.1.

The computed weight vectors are in $\Delta_{|\Gamma|}$. As we do not necessarily want to impose such a restriction, we use another trick, introduced by Kivinen & Warmuth (1997) and presented in Appendix B.2. It extends the class of comparison from the $|\Gamma|$-simplex to an L_1-ball of radius α denoted by $\mathcal{B}_\alpha \stackrel{\text{def}}{=} \left\{\mathbf{u}^\gamma \in \mathbb{R}^{|\Gamma|} \,\middle|\, \|\mathbf{u}\|_1 = \sum_{i\in\Gamma} |u^{\gamma,i}| \leqslant \alpha\right\}$. The aim is then to come close to the

cumulative error $\min_{\mathbf{u}^\gamma \in \mathcal{B}_\alpha} \sum_{t=1}^{T} \big(y_t^\gamma - \ \mathbf{u}^\gamma \cdot \ \mathbf{x}_t\big)^2$.

With the same notation as in the previous paragraph, for any node $\gamma \in \Gamma$, Theorem 5 of Gaillard et al. (2014) provides the following regret bound on the ML-Pol algorithm used in the experiments:

$$R_T^\gamma(\Delta_{|\Gamma|}) \leqslant E\sqrt{|\Gamma|(T+1)\big(1+\ln(1+T)\big)}. \tag{14}$$

With $E = 4C^2$ and Theorem 1, we obtain an upper bound on the regret $R_T(\Delta_{|\Gamma|})$, which is also of order $\sqrt{T}$ (up to poly-logarithmic terms):

$$R_T(\Delta_{|\Gamma|}) \leqslant 4C^2|\Gamma|\sqrt{|\Gamma|(T+1)\big(1+\ln(1+T)\big)} = \mathcal{O}\Big(|\Gamma|^{3/2}\sqrt{T\ln T}\Big). \tag{15}$$

5. Experiments

Our application relies on electricity consumption data of a large number of households to which we have added meteorological data. The regions of the households are also provided. The full data set is presented in Subsection 5.1. From these temporal and non-temporal data, we dispatch the households into two segmentations: the first one is based on the household location information; the second is behavioral and relies on the method presented in Subsection 5.2. We describe the experiments and analyze the results in Subsections 5.3 and 5.4.

5.1. The Underlying Real Data Set

The project "*Energy Demand Research Project*[1]", managed by Ofgem on behalf of the UK Government, was launched in late 2007 across Great Britain (see AECOM, 2018 and Schellong, 2011). Power consumptions of approximately 18,000 households with smart-type meters were collected at half-hourly intervals for about two years. The Region (the initial data set provides the level-4 NUTS[2] codes but we consider larger subdivisions – from 150,000 to 800,000 inhabitants – and associate each household with its level-3 code) is associated with each household. In a data cleaning step, we removed households with more than 5 missing consumption records over the period April 20, 2009 to July 31, 2010 (around $1,600$ households are thus kept) – the remaining missing

[1] https://www.ofgem.gov.uk/gas/retail-market/metering/transition-smart-meters/energy-demand-research-project

[2] *Nomenclature des Unités Territoriales Statistiques* (nomenclature of territorial units for statistics)

consumption data points are imputed by a linear interpolation. The final data set then contains the electrical consumption records of the 1,545 remaining households over the period from April 20, 2009 to July 31, 2010 – Ben Taieb et al. (2017), who used the same data, performed similiar pre-processing in their experiments. From now on, we will denote by $\mathcal{I}$ the set of households and by $(y_{i,t})_{1-T_0 \leqslant t \leqslant T}$ the time series of the half-hourly power consumption of the $i \in \mathcal{I}$ household. Finally, we added the temperature, visibility and humidity for each region from the NOAA[3] data: we selected a weather station (with records available over the considered period) in each region and linearly interpolated the meteorological data to get 48 measurements per day (compared to 8 initially). Considering an exponential smoothed temperature, that models the thermal inertia of buildings, is likely to improve forecasts (see among others, Taylor, 2003 and Goude, Nedellec & Kong, 2014), so we create the a-exponential smoothing of the temperature $\bar{\tau}_t^\gamma \overset{\text{def}}{=} a\bar{\tau}_{t-1}^\gamma + (1-a)\tau_t^\gamma$, where $a \in [0,1]$ – after testing several values and evaluating their performance to predict consumption on a training set, we set $a = 0.999$. For a time step t, we also introduce calendar variables: the day of the week d_t (equal to 1 for Monday, 2 for Tuesday, etc.), the half-hour of the day $h_t \in \{1, ..., 48\}$ and the position in the year $\rho_t \in [0,1]$, which takes linear values between $\rho_t = 0$ on January 1st at 00:00 and $\rho_t = 1$ on December the 31st at 23:59. Table 1 sums up the available variables of our data set and gives their range.

5.2. Behavioral Segmentation of the Households

This section briefly presents the clustering approach we used to segment the households according to their consumption behavior.

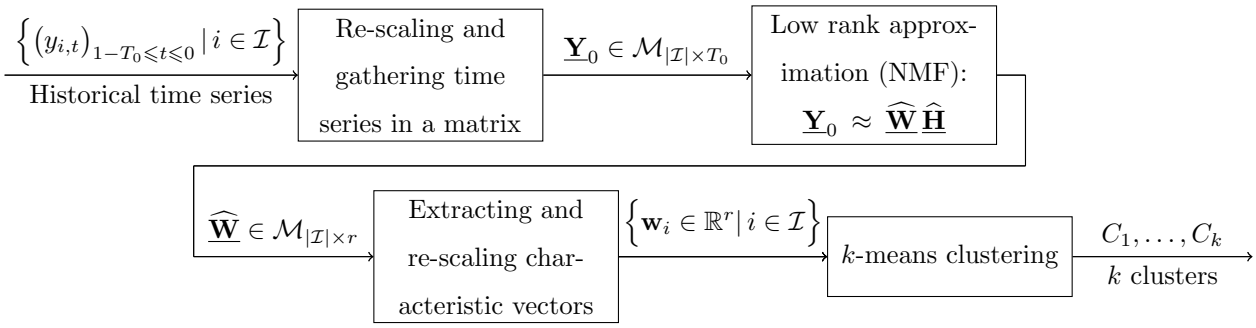
The method relies on an historical individual time series of household power consumption (April 20, 2009 to April 20, 2010). We propose a method to extract from these time series a low number – denoted by r – of combined household characteristics and to use them to build relevant segmentation. More precisely, the $|\mathcal{I}|$ historical times series $\left(y_{i,t}\right)_{1-T_0 \leqslant t \leqslant 0}$ are firstly re-scaled and gathered into a matrix $\underline{\mathbf{Y}}_0 \in \mathcal{M}_{|\mathcal{I}| \times T_0}$. We then reduce the dimension of data with a non-negative matrix factorization (NMF): we approximate $\underline{\mathbf{Y}}_0$ by $\widehat{\underline{\mathbf{W}}}\,\widehat{\underline{\mathbf{H}}}$, where $\widehat{\underline{\mathbf{W}}}$ and $\widehat{\underline{\mathbf{H}}}$ are $|\mathcal{I}| \times r$ and $r \times T_0$-non-negative matrices, respectively. As soon as this approximation is good enough, line i of the matrix $\widehat{\underline{\mathbf{W}}}$ is sufficient to reconstruct the historical time series of household i (with the knowledge of matrix $\widehat{\underline{\mathbf{H}}}$ - which is not used for the segmentation). Thus, we assign, to each household, r characteristics: the lines of $\widehat{\underline{\mathbf{W}}}$. After a re-scaling step – to give the same

[3]National Oceanic and Atmospheric Administration, https://www.noaa.gov/

Variable	Description	Range / Value
Region	UK NUTS of level 3	UK- H23, -J33, -L15, -L16, -L21, -M21, or -M27
Temperature	Air temperature	From $-20°$ to $30°$
Visibility	Air visibility	From 0 to 10 (integer)
Humidity	Air humidity percentage	From 0% to 100%
Date	Current time	From April 20, 2009 to July 31, 2010 (half-hourly)
Consumption	Power consumption	From 0.001 to 900 kWh
Half-hour	Half-hour of the day	From 1 to 48 (integer)
Day	Day off the week	From 1 (Monday) to 7 (Sunday) (integer)
Position in the year	Linear values	From 0 (Jan 1, 00:00) to 1 (Dec 31, 23:59)
Smoothed temperature	Smoothed air temperature	From $-20°$ to $30°$

Table 1: Summary of the variables provided and created for each household of the data set.

importance to each of those characteristics – we get the r-vectors $(\mathbf{w}_i)_{i\in\mathcal{I}}$. With this low-dimension representation of households in $\mathbb{R}^r$, we use the k-means clustering algorithm in $\mathbb{R}^r$ to provide the k clusters $C_1, \ldots, C_k$. These four steps are summed up in the following scheme and fully detailed in Appendix C.1.



5.3. Experiment Design

Thanks to the above method, we established a partition of the household set $\mathcal{I}$. A second segmentation is suggested by the data: the one which consists in grouping households according to the "Region" information. Once these two segmentation have been defined, we can consider two crossed hierarchies (Example 2).

We divide the data set into training data: one-year of historical data (from April 20, 2009 to April 19, 2010) – used for the behavioral segmentation, benchmark forecast generation method

training, and standardization – and testing data. As aggregation algorithms start from scratch, they work poorly during the first rounds. We therefore withdraw the first 10 days of testing data from the performance evaluation period. So, April 20, 2010 to April 30, 2010 is left for initializing aggregation algorithms and the hyper-parameters calibration and our methods are then tested during the last three months (from May 1, 2010 to July 31, 2010). We summarize in Table 2 the range of dates for each step of the procedure.

Underlying Hierarchies. As detailed in Section 2, we aim to forecast a set of power consumption time series $\left\{(y_t^\gamma)_{t>0}, \gamma \in \Gamma\right\}$ connected to each other by some summation constraints. These constraints are represented by one (or more) tree(s) and Γ denotes the set of its (or their) nodes. We refer to Example 1 if we consider a single segmentation ("Region" or "Behavior") and to Example 2 for two crossed segmentations ("Region" and "Behavior"). The partition $(A_1, \dots A_N)$ refers to segmentation "Region" and $(C_1, \dots C_k)$ to the behavioral one. We recall that we denote the average power consumption of a group of households $\gamma \subset \mathcal{I}$ by $y_t^\gamma \stackrel{\text{def}}{=} \sum_{i\in\gamma} y_{i,t}$. Considering a single segmentation $(A_n)_{1\leqslant n\leqslant N}$ or $(C_\ell)_{1\leqslant \ell\leqslant k}$ of $\mathcal{I}$, we want to forecast the consumption of each cluster D_n or C_ℓ , and also the global consumption (namely, the one for $\gamma = \mathcal{I}$). Thus, we set

$$\Gamma = \{A_n\}_{1\leqslant n\leqslant N} \cup \{\mathcal{I}\} \quad or \quad \Gamma = \{C_\ell\}_{1\leqslant \ell\leqslant k} \cup \{\mathcal{I}\}$$

and the associated time series respect the hierarchy of Figure 1 – where y^{Tot} refers to the time series associated with $\mathcal{I}$ and $y^1, y^2, \dots$ with the ones of clusters $C_1, C_2, \dots$ or $A_1, A_2, \dots$. If we now consider both partitions at the same time; we would like to forecast the global consumption ($\gamma = \mathcal{I}$), the consumption associated with each region ($\gamma = A_n$, for $n = 1, \dots, N$) and with each cluster ($\gamma = C_\ell$, for $\ell = 1, \dots, k$) but also the power consumption of cluster C_1 in region A_1 ($\gamma = C_1 \cap A_1$), of cluster C_1 in region A_2 ($\gamma = C_1 \cap A_2$), and so on. Thus,we consider the set of nodes

$$\Gamma = \{C_\ell \cap A_n\}_{1\leqslant \ell\leqslant k,\, 1\leqslant n\leqslant N} \cup \{C_\ell\}_{1\leqslant \ell\leqslant k} \cup \{A_n\}_{1\leqslant n\leqslant N} \cup \{\mathcal{I}\}. \tag{16}$$

The hierarchy associated with such crossed segmentations is represented in Figure 2 (with $N_1 = k$ and $N_2 = N$) – where the global consumption, associated with $\mathcal{I}$, is denoted by y^{Tot}, the one of cluster C_ℓ by $y^{\ell,\cdot}$, the one of region A_n, by $y^{\cdot,n}$ and where $y^{\ell,n}$ refers to the local consumption of $C_\ell \cap A_n$. Whatever the hierarchy considered, we aim to forecast the aggregated consumption of the households for each level $\gamma \in \Gamma$, namely the time series $\left(\sum_{i\in\gamma} y_{i,t}\right)_{\gamma\in\Gamma}$.

	Start date	End date
Behavioral Segmentation		
Benchmark Generation Model Training	April 20, 2009	April 19, 2010
Benchmarks and Observations Standardization		
Initialization of the Aggregation	April 20, 2010	April 30, 2010
Model Evaluation	May 1, 2010	July 31, 2010

Table 2: Date range for the steps of the proposed method

Meteorological Data of any Set of Households. The method used for benchmark forecast creation will implicitly assume that meteorological data are available. We recall that we collected meteorological data for each of the N regions. Thus when $\gamma \in \Gamma$ refers to one of these regions, we can directly apply the benchmark forecast generation methods. However, if node γ groups households from different regions, these data are not directly available and one may even wonder what they should correspond to. We take convex combinations of regional meteorological data, in proportions corresponding to the locations of the households. More precisely, for each meteorological variable (temperature, visibility or humidity), we built the meteorological variable of γ as a convex combination of the N meteorological variables of the N regions. The weight associated with region n corresponds to the proportion of this region in γ, in terms of contribution to the consumption – this contribution is determined from historical data.

Benchmark Forecast Creation. Generalized additive models (see the monograph of Wood, 2006 for an in-depth presentation) are effective semi-parametric approaches to forecast electricity consumption (see, among others, Goude et al., 2014 and Gaillard et al., 2016) which model the power demand as a sum of independent exogenous (possibly non-linear) variable effects. In our experiments, for a node $\gamma \in \Gamma$, we take into account some local meteorological variables and some calendar variables and consider a generalized additive model for the power consumption. This is fully detailed in Appendix C.2. We estimate $|\Gamma|$ generalized additive models using available explanatory variables to generate benchmark forecasts $\mathbf{x}_t$. Each of them is trained on a year of historical data (from April 20, 2009 to April 20, 2010). Then, forecasts are computed on the period April 20, 2010 to July 31, 2010.

Observations and Benchmarks Standardization. Once the benchmarks are computed, they are standardized using the protocol presented in Appendix A. We assess the quality of the standardization for one given configuration, namely "Region + Behavior", with benchmarks generated by the general additive model (this configuration, which refers to the two crossed segmentations "Region" and "Behavior") in Appendix C.3.

Calibration of Hyper-Parameters. Once benchmarks and observations are standardized, we run ML-Pol algorithm on the $|\Gamma|$ nodes, in parallel with the same hyper-parameter: we need to set α, the radius of the $L1$-ball (see Algorithm 2 of Appendix B.2). We optimize its choice by grid search, which is simply an exhaustive search in a specified finite subset G of the hyper-parameter space. This optimization is performed sequentially: such online calibration has shown good performance in power consumption forecasting (see, for example, Devaine et al., 2013).

On an Operational Constraint: Half-Hourly Predictions with One-Day-Delayed Observations. In this paragraph, we highlight the differences between the theoretical and the experimental settings and how these changes affect the regret bound. We aim to forecast power consumptions at half-hourly intervals. Meta-algorithm 1 uses historical time series values $\mathbf{y}_{1:t-1}$ to forecast at a time step t. We thus assume that very recent past observations, up to half an hour ago, would be available; and it is not realistic at all. Indeed, there are some operational constraints on the power network and on meters that make it difficult to get the data immediately: it is common to obtain load records with a delay of a few hours or even a few days. Although this delay is becoming shorter with the deployment of smart meters and the evolution of grids, we cannot consider we have access to the consumption of the previous half-hour. To take into account these operational constraints and to carry out experiments under practical conditions, we make the classic assumption that we have access to consumptions with a delay of 24 hours (see among others Fan & Hyndman, 2011 and Gaillard et al., 2016). As now, only past observations $\mathbf{y}_{1:t-48}$ are available at a time step t, we adapt the previous method a bit.

This delay is especially problematic for online learning (in our experiments, benchmark forecasts are generated offline with models trained on historical data). Indeed, in the aggregation step of our method, we assume to observe, for each node γ and at each time step t, the consumption y_{t-1}^{γ} and that is not possible anymore. To deal with this issue we initially considered two solutions. In our first approach, for any $\gamma \in \Gamma$, the time series (y_t^{γ}) is divided into 48 time series with daily time

steps. Then, 48 aggregations are done in parallel and, as $t-1$ now refers to the previous day, there is no more delay issue. The 48 series are then collected to reconstruct a time series at half-hour time step. For any set $\mathcal{D} \in \mathbb{R}^{|\Gamma|}$, the regret of the global aggregation $R_T^\gamma(\mathcal{D})$ is simply the sum of the 48 regrets – that refer to the 48 aggregation run in parallel on the 48 daily time series – denoted by $\left(R_{T/48}^{\gamma,h}(\mathcal{D})\right)_{1\leqslant h\leqslant 48}$, so we have

$$R_T^\gamma(\mathcal{D}) = \sum_{h=1}^{48} R_{T/48}^{\gamma,h}(\mathcal{D})\,. \tag{17}$$

If we consider an aggregation algorithm that ensures a bound of the form of Assumption 1 where the bound B depends only on the horizon time – namely, $R \leqslant B$ for all h – the regret associated with the half-hourly time series (y_t^γ) satisfies:

$$R_T^\gamma(\mathcal{D}) \leqslant 48 \times B(T/48)\,. \tag{18}$$

Joulani, Gyorgy & Szepesvári (2013) provide an overview of work on online learning under delayed feedback and for our framework, which refers to full information setting with general feedback. The bound above matches their results. In a second approach, we "ignore" the delay in a sense that we apply the aggregation algorithms as if the delayed observations $\mathbf{y}_{t-48}$ were $\mathbf{y}_t$.

5.4. Results

In this subsection, we compare the four forecasting strategies detailed below by evaluating them on the testing period (May 1, 2010 to July 31, 2010) for the three segmentations: "Region", "Behavior" and 'Region + Behavior". To do so, we introduce some prediction error defined below as well as a confidence bound on this error. We recall that we aim to forecast, at each time step t, a vector of time series $\mathbf{y}_t = (y_t^\gamma)_{\gamma\in\Gamma}$. The first strategy, that we call "Benchmark", consists simply in providing the benchmarksv$\mathbf{x}_t$ as forecasts. The second one considers only the projection step and thus skips the aggregation step (we will refer to it as the "Projection" strategy), the associated forecasts are thus the projected benchmark forecasts $\Pi_{\underline{\mathbf{K}}}(\mathbf{x}_t)$. To measure the impact of the aggregation step, without projection, we also evaluate the forecasts $\widehat{\mathbf{y}}_t$ (which do not necessarily satisfy the hierarchical constraints) – this strategy is called "Aggregation". Finally, the strategy "Aggregation + Projection" provides the predictions $\widetilde{\mathbf{y}}_t = \Pi_{\underline{\mathbf{K}}}(\widehat{\mathbf{y}}_t)$. To allow for an evaluation of the accuracy of the prediction of some time series only, we define the prediction error $E_T(\Lambda)$, for some subset of nodes $\Lambda \subset \Gamma$. In the results below, this subset can be equal to Γ (to evaluate the

strategies on all the nodes), to the singleton $\{\mathcal{I}\}$ (to focus on the global consumption – namely the consumption of all the households), or to the set of leaves of the tree associated with the considered segmentation(s), denoted by Γ_0 (to evaluate the performance of local forecasts only). Note that $E_T(\Gamma)$ will correspond to $\widetilde{L}_T \times |\Gamma|$ for the "Aggregation + Projection" strategy (see Equation 1). For a node $\gamma \in \Lambda$ and a time step t, let us denote by ε_t^γ the instantaneous squared error. It corresponds to $\left(y_t^\gamma - x_t^\gamma\right)^2$ for the "Benchmark" strategy, to $\left(y_t^\gamma - \left(\Pi_{\underline{\mathbf{K}}}(\mathbf{x}_t)\right)^\gamma\right)^2$ for "Projection", to $\left(y_t^\gamma - \widehat{y}_t^\gamma\right)^2$ for "Aggregation", and to $\left(y_t^\gamma - \widetilde{y}_t^\gamma\right)^2$ for the "Aggregation + Projection" strategy. We then consider the average (over time) squared prediction error (which is cumulated over Λ):

$$E_T(\Lambda) \stackrel{\text{def}}{=} \sum_{\gamma\in\Lambda} \frac{1}{T} \sum_{t=1}^{T} \varepsilon_t^\gamma. \tag{19}$$

We associate with this error a confidence bound and present our results (see Tables 3 and 4) in the form:

$$E_T(\Lambda) \pm \frac{\sigma_T(\Lambda)}{\sqrt{T}}, \quad \text{where} \quad \sigma_T(\Lambda)^2 = \frac{1}{T} \sum_{t=1}^{T} \sum_{\gamma\in\Lambda} \left(\varepsilon_t^\gamma - E_T(\Lambda)\right)^2. \tag{20}$$

We choose the quantity $\sigma_T(\Lambda)/\sqrt{T}$ as it is reminiscent of the error margin provided by asymptotic confidence intervals on the mean of independent and identically distributed random variables. In the next paragraph, we consider the "Region + Behavior" configuration and we compute the errors and confidence bounds for the four above foresting strategies for Γ, $\{\mathcal{I}\}$ and $\Gamma_0 \stackrel{\text{def}}{=} \left\{C_\ell \cap A_n\right\}_{1\leqslant\ell\leqslant 1k\, 1\leqslant n\leqslant N}$. Finally, in the last paragraph, we focus on $\{\mathcal{I}\}$ and compare the three possible configurations "Region", "Behavior" and "Region + Behavior".

5.4.1. Assessment of the Proposed Methodology

In Table 3, for the two crossed hierarchies "Region + Behavior", the forecasting errors $E_T(\Lambda) \pm \sigma_T(\Lambda)/\sqrt{T}$ are computed for the entire set of predictions (namely, for $\Lambda = \Gamma$), for the global consumption forecast only ($\Lambda = \{\mathcal{I}\}$) and for the local forecasts ($\Lambda = \Gamma_0$).

Theorem 1 was obtained for $\Lambda = \Gamma$, and as the theory guarantees, projection (with or without an aggregation step) always improves the forecasts. The projection step without aggregation leads to a decrease of prediction error of around 1%. The aggregation step leads to an improvement of almost 3%; finally, in addition to ensuring that forecasts respect hierarchical constraints, the "Aggregation + Projection" strategy reaches the best performance.

	$\mathbf{E_T}(\mathbf{\Gamma}) \pm \frac{\sigma_\mathbf{T}(\mathbf{\Gamma})}{\sqrt{\mathbf{T}}}$	$\mathbf{E_T}(\{\mathcal{I}\}) \pm \frac{\sigma_\mathbf{T}(\{\mathcal{I}\})}{\sqrt{\mathbf{T}}}$	$\mathbf{E_T}(\mathbf{\Gamma_0}) \pm \frac{\sigma_\mathbf{T}(\mathbf{\Gamma_0})}{\sqrt{\mathbf{T}}}$
Benchmark	455.5 ± 1.1	205.8 ± 9.3	66.3 ± 0.1
Projection	450.7 ± 1.1	200.8 ± 9.2	66.3 ± 0.1
Aggregation	397.9 ± 1.0	172.0 ± 8.6	61.2 ± 0.1
Aggregation + Projection	396.0 ± 1.0	170.3 ± 8.5	61.1 ± 0.1

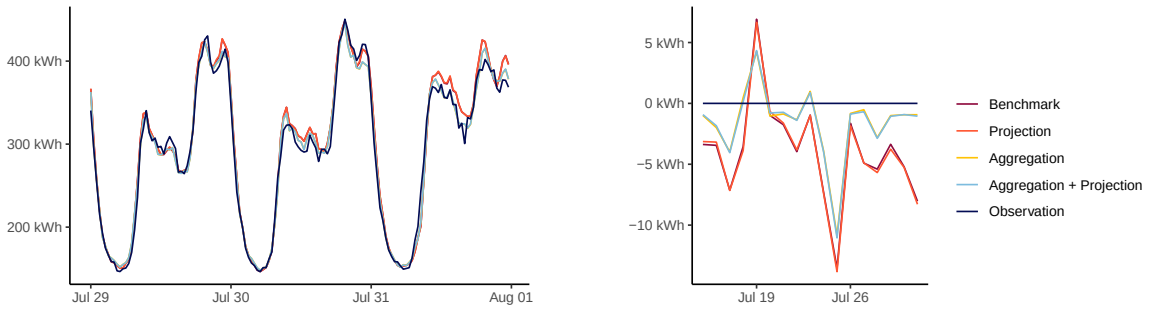
Table 3: $E_T(\Gamma) \pm \sigma_T(\Gamma)/\sqrt{T}$ (left – see Equation 20), $E_T(\{\mathcal{I}\}) \pm \sigma_T(\{\mathcal{I}\})/\sqrt{T}$ (middle), $E_T(\Gamma_0) \pm \sigma_T(\Gamma_0)/\sqrt{T}$ (right) (see Equation 20) for "Region + Behavior" clustering for the four strategies defined in Subsection 5.4 ("Benchmark", "Projection", "Agregation" and "Aggregation + Projection"). $E_T(\Gamma)$ corresponds to $\widetilde{L}_T \times |\Gamma|$ for the "Aggregation + Projection" strategy. The prediction error $E_T(\{\mathcal{I}\})$ corresponds to the mean squared error (over the testing period) of the global consumption and $E_T(\Gamma_0)$ corresponds to a prediction error associated with local consumption forecasts. The dark gray area corresponds to the best prediction error of the column.

Even though theoretical guarantees are only ensured for errors summed over all nodes, the impact of our methods on global consumption predictions and on local predictions (*i.e.*, predictions at leaves) is positive. Indeed, our strategy "Aggregation + Projection" outperforms the three strategies "Benchmark", "Aggregation" and "Projection" for both global and local power consumptions.

Finally, Figure 3 represents the global power consumption on the three last day of the testing period and the daily average signed error on the last week for the four forecasts. The distributions of the daily mean squared errors for these strategies are represented in Figure 4. We draw the same conclusions for the daily prediction errors as for the average error on the entire test period (three months): aggregation greatly improves the forecasts, projection does too, but to a lesser extent. The box plots show that the variance of the error also decreases after the aggregation step.

5.4.2. Interest of the Segmentation

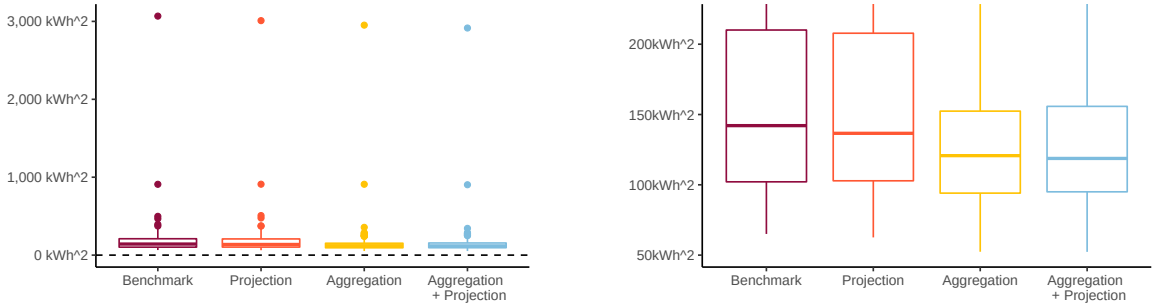
We now assess the impact of household segmentation ("Region", "Behavior" or "Region + Behavior") on the quality of our predictions. As the groups change according to which segmentations are or are not taken into account, errors related to Γ or Γ_0 can not be compared from a segmentation to another. We thus focus here on the global consumption (namely, we compute errors related to $\{\mathcal{I}\}$). We compare our methods to a naive bottom-up strategy: at each time step t, we forecast the global consumption $y_t^{\{\mathcal{I}\}}$ with the sum of local consumptions $\sum_{\gamma \in \Gamma_0} x_t^\gamma$ – in lieu of the benchmark predictions $x_t^{\{\mathcal{I}\}}$. Table 4 contains the prediction errors and the confidence bounds for



(a) Forecasts at half-hour intervals on three days.

(b) Daily average signed errors on a week.

Figure 3: Forecasts and errors associated with the four strategies "Benchmark", "Projection", "Aggregation" and "Aggregation + Projection" and observations of global consumption ($\gamma = \mathcal{I}$) at the end of the test period.



(a) Original boxplots.

(b) Boxplots trimmed at 220 kWh^2.

Figure 4: Distribution over the test period of daily mean squared error of global consumption for the four strategies "Benchmark", "Projection", "Aggregation" and "Aggregation + Projection".

Clustering	Benchmark	Bottom-up	Projection	Aggregation	Aggregation + Projection
Region	205.8 ± 9.3	189.9 ± 8.3	201.3 ± 9.1	187.8 ± 8.4	186.7 ± 8.4
Behavior	—	208.4 ± 9.6	205.2 ± 9.3	179.3 ± 8.4	179.3 ± 8.4
Region + Behavior	—	$\mathbf{201.0 \pm 8.5}$	$\mathbf{200.8 \pm 9.2}$	$\mathbf{172.0 \pm 8.6}$	$\mathbf{170.3 \pm 8.5}$

Table 4: $E_T(\{\mathcal{I}\}) \pm \sigma_T(\{\mathcal{I}\})/\sqrt{T}$ (see Equation 20) for the fives strategies defined in Subsection 5.4 ("Benchmark", "Bottom-up, "Projection", "Agregation" and "Aggregation + Projection"), with benchmark predictions ($x_t^{\{\mathcal{I}\}}$ that are the same for all clusterings) made with General Additive Models and aggregated with ML-Pol algorithm, for the three segmentations ("Region", "Behavior" and "Region + Behavior"). The prediction error $E_T(\{\mathcal{I}\})$ corresponds to the mean squared error (over the testing period) of the global consumption. The dark gray area corresponds to the best prediction error of the table and the light gray area to the best one, for a given strategy.

the five strategies and and the three configurations "Region", "Behavior" and "Region + Behavior". For the "Bottom-up" strategy, the geographical segmentation "Region" provides the lowest prediction error, that is much better than the one of benchmark forecasts. For all other strategies, it is preferable to consider the "Region + Behavior" configuration. Therefore, using information on the locality and the behavior of households is helpful to forecast global power consumption. and the strategy "aggregation + projection" strategy reaches the lowest error.

6. Conclusion

We proposed a three-step approach to forecasting electricity consumption time series at different levels of household aggregation and linked by hierarchical constraints. After generating benchmark forecasts using generalized additive models, our method aggregates them with the algorithm 'polynomially weighted average forecaster with multiple learning rates'. Finally, the forecasts are projected onto a coherent subspace to ensure that the final forecasts satisfy the hierarchical constraints. A theoretical result ensures, via a regret bound, that this approach improves the average root mean square error of the forecasts.

We tested our method on household electricity consumption data collected in the UK as part of the '*Energy Demand Research Project*'. Experimental results suggest that the successive steps of aggregation and projection improve forecasts overall (and fortunately as the theoretical result shows), but even better, that both the forecast of global consumption and local forecasts are also improved. Experiments have also shown that the segmentation considered (regional, behavioral, or both) matters. Specifically, we have seen that the use of forecasts of groups of households segmented according to their region (with their own weather conditions) on the one hand and their behavior on the other improves the global power consumption forecast.

References

AECOM (2018). Energy demand research project: early smart meter trials, 2007-2010, .

Amat, C., Michalski, T., & Stoltz, G. (2018). Fundamentals and exchange rate forecastability with simple machine learning methods. *Journal of International Money and Finance*, *88*, 1–24.

Auder, B., Cugliari, J., Goude, Y., & Poggi, J.-M. (2018). Scalable clustering of individual electrical curves for profiling and bottom-up forecasting. *Energies*, *11*, 1893.

Ben Taieb, S., Taylor, J. W., & Hyndman, R. J. (2017). Coherent probabilistic forecasts for hierarchical time series. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70* (pp. 3348–3357). JMLR. org.

Ben Taieb, S., Taylor, J. W., & Hyndman, R. J. (2020). Hierarchical probabilistic forecasting of electricity demand with smart meter data. *Journal of the American Statistical Association*, (pp. 1–17).

Cesa-Bianchi, N., & Lugosi, G. (2006). *Prediction, Learning, and Games*. Cambridge University Press.

Cover, T. M. (1991). Universal Portfolios. *Mathematical Finance*, *1*, 1–29.

Deswarte, R., Gervais, V., Stoltz, G., & Da Veiga, S. (2018). Sequential model aggregation for production forecasting, . Working paper or preprint.

Devaine, M., Gaillard, P., Goude, Y., & Stoltz, G. (2013). Forecasting electricity consumption by aggregating specialized experts. *Machine Learning*, *90*, 231–260.

Dunn, D. M., Williams, W. H., & DeChaine, T. L. (1976). Aggregate versus subaggregate models in local area forecasting. *Journal of the American Statistical Association*, *71*, 68–71.

Fan, S., & Hyndman, R. J. (2011). Short-term load forecasting based on a semi-parametric additive model. *IEEE Transactions on Power Systems*, *27*, 134–141.

Gaillard, P. (2015). *Contributions to online robust aggregation : work on the approximation error and on probabilistic forecasting. Applications to forecasting for energy markets.*. Theses Université Paris Sud - Paris XI. `https://tel.archives-ouvertes.fr/tel-01250027`.

Gaillard, P., Gerchinovitz, S., Huard, M., & Stoltz, G. (2019). Uniform regret bounds over $\mathbb{R}^d$ for the sequential linear regression problem with the square loss. In A. Garivier, & S. Kale (Eds.), *Proceedings of the 30th International Conference on Algorithmic Learning Theory* (pp. 404–432). Chicago, Illinois: PMLR volume 98 of *Proceedings of Machine Learning Research*.

Gaillard, P., Goude, Y., & Nedellec, R. (2016). Additive models and robust aggregation for GEFCom2014 probabilistic electric load and electricity price forecasting. *International Journal of Forecasting*, *32*, 1038–1050.

Gaillard, P., Stoltz, G., & van Erven, T. (2014). A second-order bound with excess losses. In M. F. Balcan, V. Feldman, & C. Szepesvári (Eds.), *Proceedings of the 27th Conference on Learning Theory* (pp. 176–196). Barcelona, Spain: PMLR volume 35 of *Proceedings of Machine Learning Research*.

Goehry, B., Goude, Y., Massart, P., & Poggi, J. (2020). Aggregation of multi-scale experts for bottom-up load forecasting. *IEEE Transactions on Smart Grid*, *11*, 1895–1904.

Goude, Y., Nedellec, R., & Kong, N. (2014). Local short and middle term electricity load forecasting with semi-parametric additive models. *IEEE Transactions on Smart Grid*, *5*, 440–446.

Gross, C. W., & Sohl, J. E. (1990). Disaggregation methods to expedite product line forecasting. *Journal of Forecasting*, *9*, 233–254.

Hyndman, R. J., Ahmed, R. A., Athanasopoulos, G., & Shang, H. L. (2011). Optimal combination forecasts for hierarchical time series. *Computational Statistics and Data Analysis*, *55*, 2579–2589.

Joulani, P., Gyorgy, A., & Szepesvári, C. (2013). Online learning under delayed feedback. In *International Conference on Machine Learning* (pp. 1453–1461).

Kivinen, J., & Warmuth, M. K. (1997). Exponentiated gradient versus gradient descent for linear predictors. *Infor-*

mation and Computation, *132*, 1–63.

Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, *401*, 788.

Littlestone, N., & Warmuth, M. K. (1994). The weighted majority algorithm. *Information and computation*, *108*, 212–261.

MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability* (pp. 281–297). Oakland, CA, USA volume 1.

Mallet, V., Stoltz, G., & Mauricette, B. (2009). Ozone ensemble forecast with machine learning algorithms. *Journal of Geophysical Research: Atmospheres*, *114*.

Paatero, P., & Tapper, U. (1994). Positive matrix factorization: a non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, *5*, 111–126.

Panagiotelis, A., Athanasopoulos, G., Gamakumara, P., & Hyndman, R. J. (2020). Forecast reconciliation: A geometric view with new insights on bias correction. *International Journal of Forecasting*, .

Schellong, W. (2011). Energy demand analysis and forecast. *Energy Management Systems*, (pp. 101–120).

Shlifer, E., & Wolff, R. W. (1979). Aggregation and proration in forecasting. *Management Science*, *25*, 594–603.

Taylor, J. W. (2003). Short-term electricity demand forecasting using double seasonal exponential smoothing. *Journal of the Operational Research Society*, *54*, 799–805.

Van Erven, T., & Cugliari, J. (2015). Game-theoretically optimal reconciliation of contemporaneous hierarchical time series forecasts. In *Modeling and Stochastic Learning for Forecasting in High Dimensions* (pp. 297–317). Springer.

Vovk, V. G. (1990). Aggregating strategies. In *Proceedings of the Third Annual Workshop on Computational Learning Theory* COLT '90 (pp. 371–386). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Wickramasuriya, S. L., Athanasopoulos, G., & Hyndman, R. J. (2019). Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization. *Journal of the American Statistical Association*, *114*, 804–819.

Wood, S. (2006). *Generalized Additive Models: An Introduction with R*. CRC Press.

Appendix A. Standardization

In empirical machine learning, it is known that standardizing observations and benchmarks may significantly improve results, and sequential learning is no exception (see Gaillard, Gerchinovitz, Huard & Stoltz, 2019). In addition, standardization makes the calibration of the parameters of the algorithm common to all the nodes, namely for each aggregation algorithm $\mathcal{A}^\gamma$, we choose the hyper-parameters $\mathbf{s}_0^\gamma = \breve{\mathbf{s}}_0$. We can do so, because thanks to the preprocessing below, benchmarks and observations will be of the same order. Let us fix $\gamma \in \Gamma$ and $t > 0$. We consider the following transformations, relying on statistics S^γ and $\breve{\underline{\mathbf{E}}}$ computed on T_0 historical time steps:

$$y_t^\gamma \to \breve{y}_t^\gamma \stackrel{\text{def}}{=} \frac{y_t^\gamma - x_t^\gamma}{S^\gamma} \qquad \text{Observations tranform} \tag{A.1}$$

$$\mathbf{x}_t \to \breve{\mathbf{x}}_t \stackrel{\text{def}}{=} \breve{\underline{\mathbf{E}}}\mathbf{x}_t \qquad \text{Benchmarks transform} \tag{A.2}$$

$$\text{with} \quad S^\gamma = \max_{1-T_0 \leqslant t \leqslant 0} |y_t^\gamma - x_t^\gamma| \qquad \text{and} \qquad \breve{\underline{\mathbf{E}}} \stackrel{\text{def}}{=} \left(\frac{1}{T_0} \sum_{t=1-T_0}^{0} \mathbf{x}_t \, \mathbf{x}_t^{\mathrm{T}} \right)^{-1/2} . \tag{A.3}$$

We thus assume that the Gram matrix $\frac{1}{T_0} \sum_{t=1-T_0}^{0} \mathbf{x}_t \mathbf{x}_t^{\mathrm{T}}$ is invertible, which is a rather reasonable assumption as long as T_0 is large enough and the basic forecasts are generated independently of each other so that they are not fully correlated. Our standardization process differs from the usual methods but it provides the theoretical guaranties set out below. Furthermore, it makes sense for the following reasons. Fixing $\gamma \in \Gamma$, when benchmarks and observations are bounded, S^γ is an estimation of a bound on $y_t^\gamma - x_t^\gamma$. The re-scaling of $(y_t^\gamma - x_t^\gamma)$ by S^γ should provide transformed observations lying in $[-1, 1]$ or some neighboring range. It also reduces and homogenizes the variances for all the nodes. A simple example may illustrate this variance reduction. For deterministic benchmarks, the variance of non-transformed observations satisfy $\mathrm{Var}\big(y_t^\gamma\big) = \mathrm{Var}\big(y_t^\gamma - x_t^\gamma\big)$. The variance of standardized observations is then divided by $\big(S^\gamma\big)^2$ and we have $\mathrm{Var}\big(\breve{y}_t^\gamma\big) = \mathrm{Var}\big(y_t^\gamma\big)/\big(S^\gamma\big)^2$. For T_0 large enough, the variance of transformed observations should be less than 1. Indeed, with high probability, the maximum of the absolute values of the random variable $(y_t^\gamma - x_t^\gamma)$ on $t = 1-T_0, \ldots, 0$ (which is S^γ), is higher than its standard deviation $\sqrt{\mathrm{Var}\big(y_t^\gamma\big)}$ and thus $\big(S^\gamma\big)^2 > \mathrm{Var}\big(y_t^\gamma\big)$. Moreover, the expectation of $(y_t^\gamma - x_t^\gamma)$ should be close to 0 as soon as the benchmark forecasts are correctly generated. Indeed, the more the benchmark forecast are relevant, the more the observations are re-centered. Concerning the benchmarks, our standardization is classic in the case of centered benchmarks. The matrix $\breve{\underline{\mathbf{E}}}^2$ would then be an estimation of the inverse of the co-variance

matrix of vectors $\mathbf{x}_t$, and the multiplication of the benchmark forecasts by $\underline{\breve{\mathbf{E}}}$ would provide transformed benchmarks whose co-variance matrix is close to the identity matrix. Here, we do not recenter observations and benchmarks with some empirical mean as it is classically done (this would be unconvenient for our regret analysis). Anyway, Subsection 5.3 provides some experimental results which confirm that our preprocessing standardizes reasonably well observations and benchmarks. Moreover, we tested classical standardization (with re-centering) on benchmarks and obtained results similar to those presented in Section 5 (but, as hinted at above, no theoretical guaranties would be associated with this classical standardization).

We run Algorithm $\mathcal{A}^\gamma$ on transformed benchmarks and observations with the initialization parameter vector $\breve{\mathbf{s}}_0$ (which does not depend on γ) and obtain a standardized prediction at node γ, denoted by $\bar{y}_t^\gamma$. Then, we transform this output to get the (non-standardized) forecast

$$\widehat{y}_t^\gamma \stackrel{\text{def}}{=} S^\gamma \bar{y}_t^\gamma + x_t^\gamma. \tag{A.4}$$

For any vector $\breve{\mathbf{u}}^\gamma \in \mathbb{R}^{|\Gamma|}$, we introduce the standardized regret associated with transformed observations and benchmarks, denoted by $\breve{R}_T^\gamma(\breve{\mathbf{u}}^\gamma)$ as:

$$\begin{aligned}
\breve{R}_T^\gamma(\breve{\mathbf{u}}^\gamma) &\stackrel{\text{def}}{=} \sum_{t=1}^T \left(\breve{y}_t^\gamma - \bar{y}_t^\gamma\right)^2 - \sum_{t=1}^T \left(\breve{y}_t^\gamma - \breve{\mathbf{u}}^\gamma \cdot \breve{\mathbf{x}}_t\right)^2 \\
&= \sum_{t=1}^T \left(\frac{y_t^\gamma - x_t^\gamma}{S^\gamma} - \frac{\widehat{y}_t^\gamma - x_t^\gamma}{S^\gamma}\right)^2 - \sum_{t=1}^T \left(\frac{y_t^\gamma - x_t^\gamma}{S^\gamma} - \breve{\mathbf{u}}^\gamma \cdot \left(\underline{\breve{\mathbf{E}}}\mathbf{x}_t\right)\right)^2 \\
&= \frac{1}{\left(S^\gamma\right)^2} \sum_{t=1}^T \left(y_t^\gamma - \widehat{y}_t^\gamma\right)^2 - \frac{1}{\left(S^\gamma\right)^2} \sum_{t=1}^T \left(y_t^\gamma - \underbrace{\left(x_t^\gamma + S^\gamma\left(\underline{\breve{\mathbf{E}}}\breve{\mathbf{u}}^\gamma\right) \cdot \mathbf{x}_t\right)}_{\mathbf{u}^\gamma \cdot \mathbf{x}_t}\right)^2.
\end{aligned} \tag{A.5}$$

In the equations above, we define $\mathbf{u}^\gamma \stackrel{\text{def}}{=} \boldsymbol{\delta}^\gamma + S^\gamma\left(\underline{\breve{\mathbf{E}}}\breve{\mathbf{u}}^\gamma\right)$ where $\boldsymbol{\delta}^\gamma \stackrel{\text{def}}{=} \left(\mathbf{1}_{\{i=\gamma\}}\right)_{i\in\Gamma}$ denotes the standard basis vector that points in the γ direction. Equivalently, $\breve{\mathbf{u}}^\gamma = \underline{\breve{\mathbf{E}}}^{-1}(\mathbf{u}^\gamma - \boldsymbol{\delta}^\gamma)/S^\gamma$, so there is a bijective correspondence between the vectors $\mathbf{u}^\gamma$ and $\breve{\mathbf{u}}^\gamma$. Therefore, by noticing that $x_t^\gamma = \boldsymbol{\delta}^\gamma \cdot \mathbf{x}_t$, the regret associated with original benchmarks and observations is related to the regret of transformed data by the following equation:

$$\breve{R}_T^\gamma(\breve{\mathbf{u}}^\gamma) = \frac{1}{\left(S^\gamma\right)^2} \sum_{t=1}^T \left(y_t^\gamma - \widehat{y}_t^\gamma\right)^2 - \frac{1}{\left(S^\gamma\right)^2} \sum_{t=1}^T \left(y_t^\gamma - \mathbf{u}^\gamma \cdot \mathbf{x}_t\right)^2 = \frac{R_T^\gamma(\mathbf{u}^\gamma)}{\left(S^\gamma\right)^2}. \tag{A.6}$$

Furthermore, as for any $\breve{\mathbf{u}} \in \mathbb{R}^{|\Gamma|}$, Assumption 1 ensures

$$\breve{R}_T^\gamma(\breve{\mathbf{u}}^\gamma) = \sum_{t=1}^{T} \left(\breve{y}_t^\gamma - \bar{y}_t^\gamma\right)^2 - \sum_{t=1}^{T} \left(\breve{y}_t^\gamma - \breve{\mathbf{u}}^{\gamma\,\mathrm{T}} \breve{\mathbf{x}}_t\right)^2 \leqslant \mathrm{B}\left(\breve{\mathbf{x}}_{1:T}, \breve{y}_{1:T}^\gamma, \breve{\mathbf{s}}_0, \breve{\mathbf{u}}^\gamma\right), \tag{A.7}$$

Combining the two previous equations yields the following proposition.

Proposition 1. *For any $\gamma \in \Gamma$ and any $\mathbf{u}^\gamma \in \mathbb{R}^{|\Gamma|}$, if Assumption 1 holds for Algorithm $\mathcal{A}^\gamma$ run on transformed observations and benchmarks $\breve{y}_{1:T}^\gamma$ and $\breve{\mathbf{x}}_{1:T}$, with the initialization parameter vector $\breve{\mathbf{s}}_0$, we have, for $T > 0$,*

$$R_T^\gamma\left(\mathbf{u}^\gamma\right) \leqslant \left(S^\gamma\right)^2 \mathrm{B}\left(\breve{\mathbf{x}}_{1:T}, \breve{y}_{1:T}^\gamma, \breve{\mathbf{s}}_0, \breve{\mathbf{u}}^\gamma\right) \quad \text{where} \quad \breve{\mathbf{u}}^\gamma = \underline{\breve{\mathbf{E}}}^{-1}(\mathbf{u}^\gamma - \boldsymbol{\delta}^\gamma)/S^\gamma. \tag{A.8}$$

On the one hand, this preprocessing justifies boundedness assumptions (13) on observations and benchmarks, that ensure some theoretical guaranties of the form requested by Assumption 1. On the other hand, this preprocessing simplifies hyper-parameters search (for the aggregation step) as we can choose the same for every series since they have similar statistics (scale and variance).

Appendix B. Aggregation Algorithm

Appendix B.1. Polynomially weighted average forecaster with Multiple Learning rates (ML-Pol)

The polynomially weighted average forecaster with Multiple Learning rates (ML-Pol) algorithm, defined in Algorithm 1, aims to provide the best convex combination of benchmark forecasts (generally called features or experts) $(x_t^i)_{i\in\Gamma}$ to predict the time series $(y_t^\gamma)_{1\leqslant t\leqslant T}$. Therefore, for any time step t, it will output a convex vector $\mathbf{u}_t^\gamma \in \Delta_\Gamma$ and the associated forecast $\widehat{y}_{t+1}^\gamma = \mathbf{u}_{t+1}^\gamma \cdot \mathbf{x}_{t+1} = \sum_{i\in\Gamma} u_{t+1}^{\gamma,i} x_{t+1}^i$. At each step, the algorithm computes the i^{th} component of $\mathbf{u}_t^\gamma$ using the cumulative regret and learning rate of benchmark i. These quantities are defined in Algorithm 1, in which, $(\mathbf{x})_+$ denotes the vector of non-negative parts of the components of $\mathbf{x}$ and E is a constant such as for any time step t and any node $i \in \Gamma$, $|2(\widehat{y}_t^\gamma - y_t^\gamma)x_t^i| \leqslant E$.

Appendix B.2. A scheme to extend the class of comparison from the simplex to an L_1-ball

For the ML-Pol algorithm, we obtained an upper bound on $R_T^\gamma(\Delta_{|\Gamma|})$. However, there is no reason for the best linear combination of benchmarks to be convex. Algorithm 2 presents a trick introduced by Kivinen & Warmuth (1997) which extends the class of comparison from the $|\Gamma|$-simplex to an L_1-ball of radius $\alpha > 0$ denoted by $\mathcal{B}_\alpha$ and provides a bound on $R_T^\gamma(\mathcal{B}_\alpha)$. Let us fix

Algorithm 1 Polynomially weighted average forecaster with Multiple Learning rates and gradient trick

aim: Predict the time series $\left(y_t^\gamma\right)_{1\leqslant t\leqslant T}$

parameter: Bound E

initialization

$\mathbf{u}_1^\gamma = (1/|\Gamma|, \ldots, 1/|\Gamma|)$

$\widehat{y}_1^\gamma = \mathbf{u}_1^\gamma \cdot \mathbf{x}_1$

$\forall\, i \in \Gamma$, $\widetilde{R}_0^{\gamma,i} = 0$ and $\eta_0^{\gamma,i} = 0$

for $t = 1, \ldots, T-1$ **do**

For each $i \in \Gamma$, update the cumulative regret of benchmark i

$$\widetilde{R}_t^{\gamma,i} = \widetilde{R}_{t-1}^{\gamma,i} + \widetilde{r}_t^{\gamma,i} \quad \text{where} \quad \widetilde{r}_t^{\gamma,i} \stackrel{\text{def}}{=} 2(\widehat{y}_t^\gamma - y_t^\gamma)(\widehat{y}_t^\gamma - x_t^i)$$

For each $i \in \Gamma$, compute the learning rate $\eta_t^{\gamma,i} = \left(E + \sum_{s=1}^t \left(\widetilde{r}_t^{\gamma,i}\right)^2\right)^{-1}$

Compute the weight vector $\mathbf{u}_{t+1}^\gamma = (u_{t+1}^{\gamma,i})_{i\in\Gamma}$ defined as

$$u_{t+1}^{\gamma,i} = \frac{\eta_t^{\gamma,i}\left(\widetilde{R}_t^{\gamma,i}\right)_+}{\sum_{j\in\Gamma} \eta_t^{\gamma,j}\left(\widetilde{R}_t^{\gamma,j}\right)_+}$$

Output prediction $\widehat{y}_{t+1}^\gamma = \mathbf{u}_{t+1}^\gamma \cdot \mathbf{x}_{t+1} = \sum_{i\in\Gamma} u_{t+1}^{\gamma,i} x_{t+1}^i$

a node $\gamma \in \Gamma$. The trick consists in transforming, at each round t, the benchmark vector $\mathbf{x}_t$ into the $2|\Gamma|$-vector $\bar{\mathbf{x}}_t = (\alpha\mathbf{x}_t | -\alpha\mathbf{x}_t)$ – where $|$ is the concatenation operator between vectors. The algorithm $\mathcal{A}^\gamma$ is then run with these new benchmarks and it outputs the weight vector $\bar{\mathbf{u}}_t^\gamma \in \Delta_{2|\Gamma|}$. Finally, a $|\Gamma|$-vector $\mathbf{u}_t^\gamma \in \mathcal{B}_\alpha$ is computed from $\bar{\mathbf{u}}_t^\gamma$ to provide the forecast $\mathbf{u}_t^\gamma \cdot \mathbf{x}_t = \bar{\mathbf{u}}_t^\gamma \cdot \bar{\mathbf{x}}_t$. We will actually see that we may associate any $|\Gamma|$-vector $\mathbf{u} \in \mathcal{B}_\alpha$ with a vector $\bar{\mathbf{u}} \in \Delta_{2|\Gamma|}$ such as $\bar{\mathbf{u}} \cdot \bar{\mathbf{x}}_t = \mathbf{u} \cdot \mathbf{x}_t$; the trick actually defines a surjection from $\Delta_{2|\Gamma|}$ to $\mathcal{B}_\alpha$. Thus, to compete against the best linear combination of benchmarks in $\mathcal{B}_\alpha$, it is enough to compete against the best convex combination of benchmarks $\bar{\mathbf{x}}_t$ in a lifted space (which we may achieve, thanks to algorithm $\mathcal{A}^\gamma$). We now give all the details on how this trick works and indicate its impact on the stated regret bounds. The following lemma introduces the surjection from $\Delta_{2|\Gamma|}$ to $\mathcal{B}_\alpha$, which is used in Algorithm 2.

Lemma 1. *For any real $\alpha > 0$, the following function ψ is a surjection from $\Delta_{2|\Gamma|}$ to $\mathcal{B}_\alpha$:*

$$\psi : \begin{array}{lcl} \Delta_{2|\Gamma|} & \longrightarrow & \mathcal{B}_\alpha \\ \bar{\mathbf{u}} = \left(\bar{\mathbf{u}}^{\mathrm{P}} \,|\, \bar{\mathbf{u}}^{\mathrm{N}}\right) & \longmapsto & \alpha(\bar{\mathbf{u}}^{\mathrm{P}} - \bar{\mathbf{u}}^{\mathrm{N}}), \end{array}$$

where the vector $\bar{\mathbf{u}} \in \Delta_{2|\Gamma|}$ is decomposed in the two $|\Gamma|$-vectors $\bar{\mathbf{u}}^{\mathrm{P}}$ and $\bar{\mathbf{u}}^{\mathrm{N}}$, which correspond to the $|\Gamma|$ first and the $|\Gamma|$ last coefficients of $\bar{\mathbf{u}}$, respectively.

Proof of Lemma 1. Denoting respectively by $(\mathbf{u})_+$ and $(\mathbf{u})_-$ the non-negative and non-positive parts of any vector $\mathbf{u}$ and by $\mathbf{1}_{|\Gamma|}$ the vector of size $|\Gamma|$ of which all coordinates are 1, we introduce the inverse function ψ^{-1}:

$$\psi^{-1} : \begin{array}{lcl} \mathcal{B}_\alpha & \longrightarrow & \Delta_{2|\Gamma|} \\ \mathbf{u} & \longmapsto & \frac{1}{\alpha}\left(\frac{\alpha - \|\mathbf{u}\|_1}{2|\Gamma|}\mathbf{1}_{|\Gamma|} + (\mathbf{u})_+ \,\middle|\, \frac{\alpha - \|\mathbf{u}\|_1}{2|\Gamma|}\mathbf{1}_{|\Gamma|} + (\mathbf{u})_- \right). \end{array}$$

First we will show that function images are in the right sets, meaning that for any $\mathbf{u} \in \mathcal{B}_\alpha$, $\psi^{-1}(\mathbf{u}) \in \Delta_{2|\Gamma|}$ and for any $\bar{\mathbf{u}} \in \Delta_{2|\Gamma|}$, $\psi(\bar{\mathbf{u}}) \in \mathcal{B}_\alpha$. Secondly, we obtain the surjectivity of ψ by proving that for any $\mathbf{u} \in \mathcal{B}_\alpha, \psi(\psi^{-1}(\mathbf{u})) = \mathbf{u}$.

Proof that for any $\mathbf{u} \in \mathcal{B}_\alpha$, $\psi^{-1}(\mathbf{u}) \in \Delta_{2|\Gamma|}$. We set $\mathbf{u} \in \mathcal{B}_\alpha$. By definition for any $i \in \Gamma$, $(u^i)_\pm \geqslant 0$ and as $\mathbf{u} \in \mathcal{B}_\alpha$, $(\alpha - \|\mathbf{u}\|_1)/(2|\Gamma|) \geqslant 0$. So, all the coefficients of $\psi^{-1}(\mathbf{u})$ are non-negative. Since $\sum_{i\in\Gamma} (u^i)_+ + (u^i)_- = \sum_{i\in\Gamma} |u^i| = \|\mathbf{u}\|_1$, the sum of the coefficients of the vector $\psi^{-1}(\mathbf{u})$ equals 1:

$$\begin{aligned} \sum_{i\in\Gamma} \left(\left(\psi^{-1}(\mathbf{u})\right)^i \right)^{\mathrm{P}} + \left(\left(\psi^{-1}(\mathbf{u})\right)^i \right)^{\mathrm{N}} &= \frac{1}{\alpha} \sum_{\gamma\in\Gamma} \left(\left(u^i\right)_+ + \left(u^i\right)_- + \frac{\alpha - \|\mathbf{u}\|_1}{|\Gamma|} \right) \\ &= \frac{1}{\alpha}\left(\|\mathbf{u}\|_1 + \alpha - \|\mathbf{u}\|_1\right) = 1. \end{aligned} \tag{B.1}$$

and thus $\bar{\mathbf{u}} = \psi(\mathbf{u}) \in \Delta_{2|\Gamma|}$.

Proof that for any $\bar{\mathbf{u}} \in \Delta_{2|\Gamma|}$, $\psi(\bar{\mathbf{u}}) \in \mathcal{B}_\alpha$. With $\bar{\mathbf{u}} = \left(\bar{\mathbf{u}}^{\mathrm{P}} \,|\, \bar{\mathbf{u}}^{\mathrm{N}}\right) \in \Delta_{2|\Gamma|}$, using that all the coefficients of $\bar{\mathbf{u}}$ are non-negative and that their sum equals 1 that is $\left\|\bar{\mathbf{u}}\right\|_1 = 1$, we get

$$\|\psi(\bar{\mathbf{u}})\|_1 \stackrel{\text{def}}{=} \left\|\alpha\bar{\mathbf{u}}^{\mathrm{P}} - \alpha\bar{\mathbf{u}}^{\mathrm{N}}\right\|_1 \leqslant \alpha\left\|\bar{\mathbf{u}}^{\mathrm{P}}\right\|_1 + \alpha\left\|\bar{\mathbf{u}}^{\mathrm{N}}\right\|_1 = \alpha\left\|\bar{\mathbf{u}}\right\|_1 = \alpha. \tag{B.2}$$

Proof that for any $\mathbf{u} \in \mathcal{B}_\alpha$, $\psi\big(\psi^{-1}(\mathbf{u})\big) = \mathbf{u}$.

$$\psi\big(\psi^{-1}(\mathbf{u})\big) = \frac{\alpha - \|\mathbf{u}\|_1}{2|\Gamma|}\mathbf{1}_{|\Gamma|} + (\mathbf{u})_+ - \frac{\alpha - \|\mathbf{u}\|_1}{2|\Gamma|}\mathbf{1}_{|\Gamma|} - (\mathbf{u})_- = \mathbf{u}\,. \tag{B.3}$$

□

By running Algorithm $\mathcal{A}^\gamma$ with transformed benchmarks $\bar{\mathbf{x}}_t \overset{\text{def}}{=} \big(\alpha \mathbf{x}_t \,|\, -\alpha \mathbf{x}_t\big)$ and parameter s_0^γ (which provides weight vectors $\bar{\mathbf{u}}_t^\gamma$), we get the bound

$$R_T^\gamma(\Delta_{2|\Gamma|}) \overset{\text{def}}{=} \sum_{t=1}^T \big(y_t^\gamma - \bar{\mathbf{u}}_t^\gamma \cdot \bar{\mathbf{x}}_t\big)^2 - \min_{\bar{\mathbf{u}}^\gamma \in \Delta_{2|\Gamma|}} \sum_{t=1}^T \big(y_t^\gamma - \bar{\mathbf{u}}^\gamma \cdot \bar{\mathbf{x}}_t\big)^2 \leqslant \mathrm{B}(\bar{\mathbf{x}}_{1:T}, y_{1:T}^\gamma, \mathbf{s}_0^\gamma, \bar{\mathbf{u}}^\gamma). \tag{B.4}$$

For any time step $t = 1, \ldots, T$, and for any $\bar{\mathbf{u}} \in \Delta_{2|\Gamma|}$, we obtain the equality of the two scalar products $\bar{\mathbf{u}} \cdot \bar{\mathbf{x}}_t$ and $\psi(\bar{\mathbf{u}}) \cdot \mathbf{x}_t$:

$$\bar{\mathbf{u}} \cdot \bar{\mathbf{x}}_t = \big(\bar{\mathbf{u}}^{\mathrm{P}} \,|\, \bar{\mathbf{u}}^{\mathrm{N}}\big) \cdot \big(\alpha \mathbf{x}_t \,|\, -\alpha \mathbf{x}_t\big) = \alpha\big(\bar{\mathbf{u}}^{\mathrm{P}} - \bar{\mathbf{u}}^{\mathrm{N}}\big) \cdot \mathbf{x}_t = \psi(\bar{\mathbf{u}}) \cdot \mathbf{x}_t \,. \tag{B.5}$$

Lemma 1 implies that for any $\mathbf{u}^\gamma \in \mathcal{B}_\alpha$, there is at least one vector $\bar{\mathbf{u}}^\gamma \in \Delta_{2|\Gamma|}$ such that $\psi(\bar{\mathbf{u}}^\gamma) = \mathbf{u}^\gamma$ and we get the equality:

$$\min_{\mathbf{u}^\gamma \in \mathcal{B}_\alpha} \sum_{t=1}^T \big(y_t^\gamma - \mathbf{u}^\gamma \cdot \mathbf{x}_t\big)^2 = \min_{\bar{\mathbf{u}}^\gamma \in \Delta_{2|\Gamma|}} \sum_{t=1}^T \big(y_t^\gamma - \psi(\bar{\mathbf{u}}^\gamma) \cdot \mathbf{x}_t\big)^2 = \min_{\bar{\mathbf{u}}^\gamma \in \Delta_{2|\Gamma|}} \sum_{t=1}^T \big(y_t^\gamma - \bar{\mathbf{u}}^\gamma \cdot \bar{\mathbf{x}}_t\big)^2 . \tag{B.6}$$

So with, for any time step $t = 1, \ldots, T$, $\mathbf{u}_t^\gamma \overset{\text{def}}{=} \psi(\bar{\mathbf{u}}_t^\gamma)$, we obtain

$$R_T^\gamma(\mathcal{B}_\alpha) \overset{\text{def}}{=} \sum_{t=1}^T \big(y_t^\gamma - \mathbf{u}_t^\gamma \cdot \mathbf{x}_t\big)^2 - \min_{\mathbf{u}^\gamma \in \mathcal{B}_\alpha} \sum_{t=1}^T \big(y_t^\gamma - \mathbf{u}^\gamma \cdot \mathbf{x}_t\big)^2 = R_T^\gamma(\Delta_{2|\Gamma|}) \,. \tag{B.7}$$

This equality provides a bound on $R_T^\gamma(\mathcal{B}_\alpha)$ when predictions are $\widehat{y}_t^\gamma = \psi^{-1}(\bar{\mathbf{u}}_t^\gamma) \cdot \mathbf{x}_t = \mathbf{u}_t^\gamma \cdot \mathbf{x}_t$ (ψ^{-1} is defined in the proof of Lemma 1). With this trick, the bound (14) is still true by replacing $|\Gamma|$ (the dimension of the benchmarks $\mathbf{x}_t$) by $2|\Gamma|$ (the dimension of the new benchmarks $\bar{\mathbf{x}}_t$) and the bound E (previously equals to $4C^2$) by $2\alpha(\alpha+1)C^2$ (the bound on the new prediction errors are calculated below):

$$R_T(\mathcal{B}_\alpha) \leqslant \, 2\,\alpha\,(\alpha+1)\,C^2\,|\Gamma| \sqrt{|\Gamma|(T+1)\big(1+\ln(1+T)\big)}. \tag{B.8}$$

The complete online algorithm leading to these bounds is summarized in Algorithm 2.

Bound on new prediction errors. Since boundedness assumptions (13) hold, the transformed benchmarks $\bar{\mathbf{x}}_t^\gamma$ are bounded by αC. Moreover, $\bar{\mathbf{u}}_t^\gamma \in \Delta_{2|\Gamma|}$ implies $\big\|\bar{\mathbf{u}}_t^\gamma\big\|_1 = 1$, so we get

$$\big|\widehat{y}_t^\gamma\big| = \big|\bar{\mathbf{u}}_t^\gamma \cdot \bar{\mathbf{x}}_t\big| \leqslant \big\|\bar{\mathbf{u}}_t^\gamma\big\|_1 \big\|\bar{\mathbf{x}}_t\big\|_\infty = \alpha C \,. \tag{B.9}$$

Moreover, as the observations are still bounded by C, we have $|y_t^\gamma - \widehat{y}_t^\gamma| \leqslant |y_t^\gamma| + |\widehat{y}_t^\gamma| \leqslant \big(\alpha+1\big)C$ and we obtain a bound on the prediction errors:

$$\big|\widetilde{\ell}_t^\gamma(\bar{\mathbf{x}}_t)\big| = \big\|2\big(\widehat{y}_t^\gamma - y_t^\gamma\big)\bar{\mathbf{x}}_t\big\|_\infty \leqslant 2\alpha(1+\alpha)C^2 \,. \tag{B.10}$$

Algorithm 2 Scheme for on-line linear regression.

input Algorithm $\mathcal{A}^\gamma$ and bound on the weight vectors $\alpha > 0$

for $t = 1, \ldots, T$ **do**

Get the benchmark vector $\mathbf{x}_t$ and denote (where $|$ is the concatenation operator between vectors)

$$\bar{\mathbf{x}}_t \stackrel{\text{def}}{=} \big(\alpha \mathbf{x}_t \,|\, -\alpha \mathbf{x}_t\big) \in \mathbb{R}^{2|\Gamma|}$$

Run algorithm $\mathcal{A}^\gamma$ on node γ with $\bar{\mathbf{x}}_t$ and get the weight vector $\bar{\mathbf{u}}_t^\gamma = \big(\bar{\mathbf{u}}_t^{\gamma+} \,|\, \bar{\mathbf{u}}_t^{\gamma-}\big)$

Output the weight vector $\mathbf{u}_t^\gamma = \alpha\big(\bar{\mathbf{u}}_t^{\gamma+} - \bar{\mathbf{u}}_t^{\gamma-}\big)$ and predicts $\widehat{y}_t^\gamma = \mathbf{u}_t^\gamma \cdot \mathbf{x}_t$

Appendix C. Experiments

Appendix C.1. Behavioral Segmentation

Re-scaling and Gathering Time Series in a Matrix. For $T_0 > 0$, we consider the $|\mathcal{I}| \times T_0$-matrix $\underline{\mathbf{Y}}_0$ which contains the re-scaled historical power consumption time series: for any $i \in \mathcal{I}$ and any $1 - T_0 \leqslant t \leqslant 0$,

$$(\underline{\mathbf{Y}}_0)_{i,t} \stackrel{\text{def}}{=} \frac{y_{i,t}}{\bar{y}_i}, \quad \text{with} \quad \bar{y}_i \stackrel{\text{def}}{=} \frac{1}{T_0} \sum_{t=1-T_0}^{0} y_{i,t}. \tag{C.1}$$

Low Rank Approximation. Since we are interested in power consumption, all the coefficients of $\underline{\mathbf{Y}}_0$ are non-negative - we will write $\underline{\mathbf{Y}}_0 \geqslant 0$ and say that this matrix is non-negative. To reduce dimension of non-negative matrices, Paatero & Tapper (1994) and Lee & Seung (1999) propose a factorization method whose distinguishing benchmark is the use of non-negativity constraints. Let us fix some integer $r \ll \min(|\mathcal{I}|, T_0)$, which will ensure a reduction of the dimension (we chose $r = 10$ in the experiments of the next subsection). The non-negative matrix factorization (NMF) approximates matrix $\underline{\mathbf{Y}}_0$ by $\underline{\mathbf{Y}}_0 \approx \underline{\mathbf{W}}^\star \underline{\mathbf{H}}^\star$, where $\underline{\mathbf{W}}^\star$ and $\underline{\mathbf{H}}^\star$ are $|\mathcal{I}| \times r$ and $r \times T_0$ non-negative matrices. They are computed by solving:

$$\big(\underline{\mathbf{W}}^\star, \underline{\mathbf{H}}^\star\big) \in \underset{\underline{\mathbf{W}}, \underline{\mathbf{H}} \geqslant 0}{\arg\min} \big\|\underline{\mathbf{Y}}_0 - \underline{\mathbf{W}}\underline{\mathbf{H}}\big\|_F^2 = \underset{\underline{\mathbf{W}}, \underline{\mathbf{H}} \geqslant 0}{\arg\min} \sum_{i,t} \Big(y_{i,t} - \big(\underline{\mathbf{W}}\underline{\mathbf{H}}\big)_{i,t}\Big)^2. \tag{C.2}$$

We use the function `NMF` of the `Python`-library `sklearn.decomposition` to approach a local minimum with a coordinate descent solver and denote by $\widehat{\underline{\mathbf{W}}}$ the approximation of $\underline{\mathbf{W}}^\star$. Thanks to the NMF, for any $i \in \mathcal{I}$, r characteristics (the i^{th} line of matrix $\widehat{\underline{\mathbf{W}}}$) are thus computed.

Extracting and Re-scaling Characteristic Vectors. To give the same impact to each of these characteristics, we re-scale the columns of $\widehat{\underline{\mathbf{W}}}$ and define, for each household i, the vector

$$\mathbf{w}_i = \left(\frac{\widehat{\underline{\mathbf{W}}}_{i,1}}{\sum_{j\in\mathcal{I}} \widehat{\underline{\mathbf{W}}}_{j,1}} , \ldots , \frac{\widehat{\underline{\mathbf{W}}}_{i,r}}{\sum_{j\in\mathcal{I}} \widehat{\underline{\mathbf{W}}}_{j,r}} \right). \tag{C.3}$$

k-Means Clustering. The k-means algorithm (introduced by MacQueen et al. (1967)) is then used on these r- vectors to cluster the households into a fixed number k of groups (which varies from 4 to 64 in our experiments). We recall below how this algorithm works. With $\{C_1, \ldots, C_k\}$ a k-clustering of set $\mathcal{I}$, for any $1 \leqslant \ell \leqslant k$, we define the center $\bar{\mathbf{w}}_\ell$ and the variance $\mathrm{Var}(C_\ell)$ of cluster C_ℓ by

$$\bar{\mathbf{w}}_\ell \stackrel{\text{def}}{=} \frac{1}{|C_\ell|} \sum_{i\in C_\ell} \mathbf{w}_i \quad \text{and} \quad \mathrm{Var}(C_\ell) \stackrel{\text{def}}{=} \frac{1}{C_\ell} \sum_{i\in C_\ell} \|\mathbf{w}_i - \bar{\mathbf{w}}_\ell\|^2. \tag{C.4}$$

In k-means clustering, each household belongs to the cluster with the nearest center. The best set of clusters, denoted by $\{C_1^\star, \ldots, C_k^\star\}$ – namely the best set of centers – is obtained by minimizing the following criterion:

$$\{C_1^\star, \ldots, C_k^\star\} \in \underset{\{C_1,\ldots,C_k\}}{\arg\min} \sum_{\ell=1}^{k} \sum_{\mathbf{w}\in C_\ell} \left\|\mathbf{w} - \bar{\mathbf{w}}_\ell\right\|^2 = \underset{\{C_1,\ldots,C_k\}}{\arg\min} \sum_{\ell=1}^{k} |C_\ell| \mathrm{Var}(C_\ell). \tag{C.5}$$

In practice, we use the use `KMeans` function of the `Python`-library `sklearn.cluster` to compute clusters.

Appendix C.2. Generation of the benchmarks

We refer to the monograph of Wood, 2006 for an exhaustive presentations of the generalized additive models, which are commonly used to forecast the power consumption (see e.g, Goude et al., 2014 and Gaillard et al., 2016). They assume that the power consumption is the sum of independent exogenous (possibly non-linear) variable effects. We describe this model using the specification we chose in our experiments. For a node $\gamma \in \Gamma$, we take into account some local meteorological variables at the half-hour time step: the temperature τ^γ and the smoothed temperature $\bar{\tau}^\gamma$, the visibility ν^γ, and the humidity κ^γ. and some calendar variables: the day of the week d_t, the half-hour of the day $h_t \in \{1, ..., 48\}$ and the position in the year $\rho_t \in [0, 1]$. As the effect of the half-hour h_t is crucial to forecast power consumption, it is often more efficient to consider a model per half-hour (see Fan & Hyndman, 2011 and Goude et al., 2014). The global

model is then the sum of 48 daily models, one for each half-hour of the day. More precisely, the considered additive model for the power consumption breaks down time by half hours:

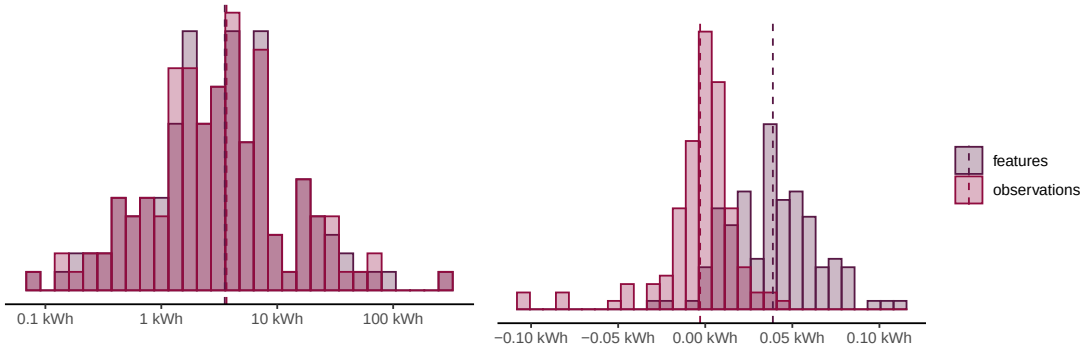
$$y_t^\gamma = \sum_{h=1}^{48} \mathbf{1}_{h_t=h} \Big[a_h^\gamma y_{t-7\times 48}^\gamma + s_{1,h}^\gamma\big(y_{t-48}^\gamma\big) + s_{\tau,h}^\gamma(\tau_t^\gamma) + s_{\bar{\tau},h}^\gamma(\bar{\tau}_t^\gamma) + s_{\nu,h}^\gamma(\nu_t^\gamma) + s_{\kappa,h}^\gamma(\kappa_t^\gamma) + \sum_{d=1}^{7} w_{d,h}^\gamma \mathbf{1}_{d_t=d} + s_{\rho,h}^\gamma(\rho_t) \Big] + \text{noise}. \tag{C.6}$$

The $s_{1,h}^\gamma$, $s_{\tau,h}^\gamma$, $s_{\bar{\tau},h}^\gamma$, $s_{\nu,h}^\gamma$, $s_{\kappa,h}^\gamma$ and $s_{\rho,h}^\gamma$ functions catch the effect of the consumption lag, the meteorological variables and of the yearly seasonality. They are cubic splines: $\mathcal{C}^2$-smooth functions made up of sections of cubic polynomials joined together at points of a grid. The coefficients a_h^γ and $w_{d,h}^\gamma$ model the influence of the consumption at D-7 and of the day of the week; we consider a linear effect for the consumption at D-7 (it achieved a better performance than a spline effect in our experiments) and as the day of the week takes only 7 values, we write its effect as a sum of indicator functions, and thus there are 7 coefficients $w_{d,h}^\gamma$. As we consider a model per half-hour, all the coefficients and splines are indexed by h. To estimate each model, we use the Penalized Iterative Re-Weighted Least Square (P-IRLS) method Wood, 2006, implemented in the `mgcv` R-package, on a training data set. At any node $\gamma \in \Gamma$, for a new round t, we then output the forecast

$$x_t^\gamma = \sum_{h=1}^{48} \mathbf{1}_{h_t=h} \Big[\widehat{a}_h^\gamma y_{t-7\times 48}^\gamma + \widehat{s}_{1,h}^\gamma\big(y_{t-48}^\gamma\big) + \widehat{s}_{\tau,h}^\gamma(\tau_t^\gamma) + \widehat{s}_{\bar{\tau},h}^\gamma(\bar{\tau}_t^\gamma) + \widehat{s}_{\nu,h}^\gamma(\nu_t^\gamma) + \widehat{s}_{\kappa,h}^\gamma(\kappa_t^\gamma) + \sum_{d=1}^{7} \widehat{w}_{d,h} \mathbf{1}_{d_t=d} + \widehat{s}_{\rho,h}^\gamma(\rho_t) \Big]. \tag{C.7}$$

Appendix C.3. Observations and benchmarks Standardization

Once above benchmarks computed, they are standardized using the protocol presented in Appendix A. We assess the quality of the standardization for the configuration "Region + Behavioral", which refers to the two crossed segmentations "Region" and ' Behavioral". As there are 7 regions, the set Γ consists of $16 \times 7 + 16 + 7 + 1 = 136$ nodes, but only 129 are non-empty. For both standardized and non-standardized observations and benchmarks, we compute, for each node $\gamma \in \Gamma$, the empirical mean and empirical standard deviation over the test period. The distributions are plotted in Figures C.5 and C.6, respectively. Since the abscissa for non-standardized data is in logarithmic scale, the mean and standard deviation of data differ a lot from a node to another. For example, the right-hand point is the global consumption ($\gamma = \mathcal{I}$), while points on the left correspond to the consumptions of small clusters. Thus, standardization centers data and decreases



(a) Non-standardized observations and benchmarks.

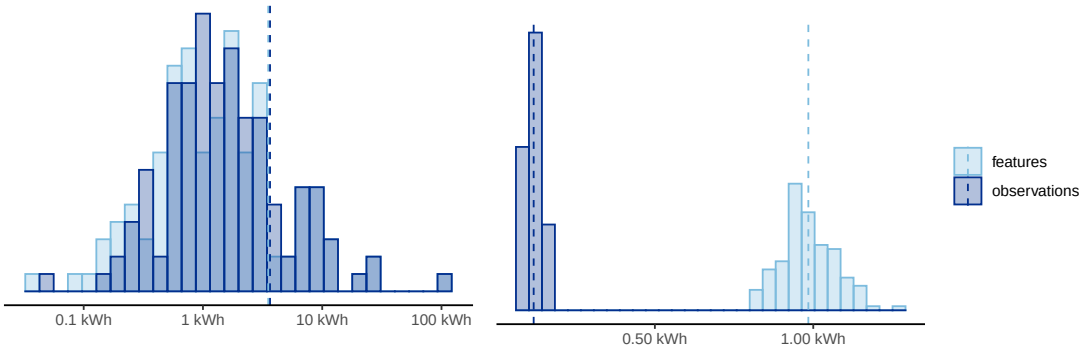
(b) Standardized observations and benchmarks.

Figure C.5: Distribution of empirical means per cluster for non-standardized and standardized observations and benchmarks.

standard deviations of observations, as desired. In addition, standard deviations of benchmarks are close to 1. Figure C.7 represents correlation matrices of the $|\Gamma|$-vectors $(\mathbf{x}_t)_{1\leqslant t\leqslant T}$ and $(\check{\mathbf{x}}_t)_{1\leqslant t\leqslant T}$, that contain the non-standardized and standardized benchmarks over the test period. This shows that our standardization process is centering, re-scaling and de-correlating benchmarks. Finally, Table C.5 gathers numerical values of the average, over $\gamma \in \Gamma$, of empirical means and standard deviations (these values are indicated by dashed vertical lines on Figures C.5 and C.6). We also compute the maximum of the absolute value of benchmarks and observations – "Bound" column of the table. This gives an empirical approximation of the boundedness constant C – see boundedness assumptions (13).

	Mean	**Bound**	**Standard deviation**
Observations	9.53	570.02	3.65
benchmarks	9.54	570.87	3.53
Standardized observations	-0.003	1.27	0.12
Standardized benchmarks	0.04	18.9	0.98

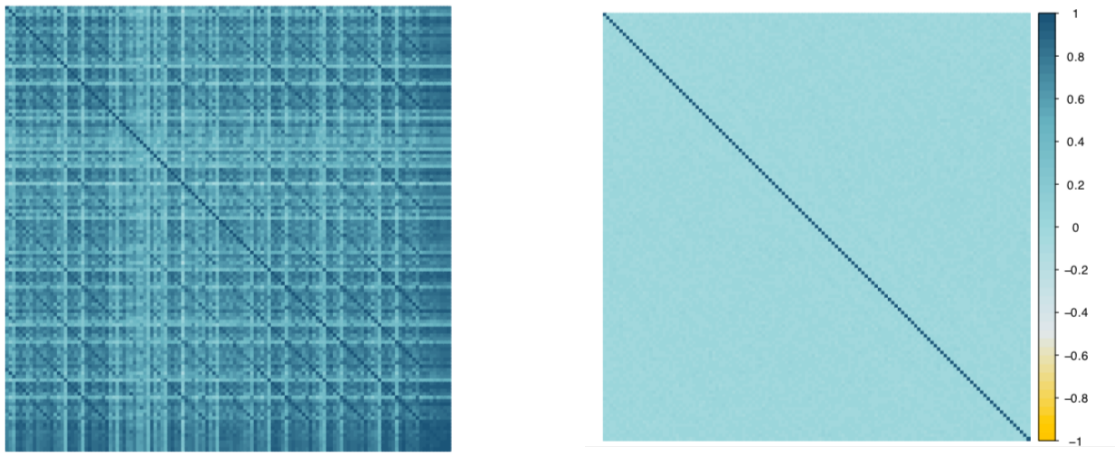
Table C.5: Mean, and maximum of absolute value and standard deviation of observations and benchmarks before and after standardization



(a) Non-standardized observations and benchmarks.

(b) Standardized observations and benchmarks.

Figure C.6: Distribution of empirical standard deviations per cluster, for non-standardized and standardized observations and benchmarks.



(a) Non-standardized benchmarks.

(b) Standardized benchmarks.

Figure C.7: Correlation matrix of non-standardized (left) and standardized (right) benchmark vectors.