Electricity Demand Forecasting by Multi-Task Learning

Jean-Baptiste Fiot, Francesco Dinuzzo

Electricity Demand Forecasting by Multi-Task Learning

Jean-Baptiste Fiot Francesco Dinuzzo
IBM Research - Ireland

***Abstract*—We explore the application of kernel-based multi-task learning techniques to forecast the demand of electricity measured on multiple lines of a distribution network. We show that recently developed output kernel learning techniques are particularly well suited to solve this problem, as they allow to flexibly model the complex seasonal effects that characterize electricity demand data, while learning and exploiting correlations between multiple demand profiles. We also demonstrate that kernels with a multiplicative structure yield superior predictive performance with respect to the widely adopted (generalized) additive models. Our study is based on residential and industrial smart meter data provided by the Irish Commission for Energy Regulation (CER).**

***Index Terms*—Electricity Demand Forecasting, Multi-Task Learning, Output Kernel Learning**

I. Introduction

Electricity cannot be stored efficiently in large quantities, therefore it is critical to ensure that the amount generated at a given time is sufficient to meet the load plus the losses while not exceeding this amount significantly. Predictive methods for accurately forecasting the demand of electricity have thus become important tools that guide planning and operation of utility companies. While electric load forecasting is a well-established, several decades old research area in engineering, new modeling problems keep appearing as technological and legislative transformations affect the power industry. With the advent of smart grids and meters, larger and richer sources of data are becoming available, making it possible to build more sophisticated models that enable more accurate billing of electricity and dynamic pricing.

A variety of tools from time series analysis, statistics, and more recently machine learning, have been employed for electricity load forecasting. For an overview on the vast body of available literature on the subject, we refer the reader to the recent book by [1]. Classical techniques include linear and non-linear regression models estimated by means of variants of least squares fitting, and various types of ARMAX models expressing the forecast as a function of previously observed values of the load and possibly other weather or social variables. Techniques inspired by Artificial Intelligence research such as expert systems, fuzzy logic, and neural networks have also been applied to load forecasting. In particular, black-box models based on neural networks have been extensively analyzed, see the influential review by [2].

In recent years, Generalized Additive Models (GAM) [3] have established themselves as state of the art tools for electricity load forecasting [4], [5], [6], due to the existence of efficient and scalable training algorithms and the interpretability of the model, which allows to clearly visualize the effect of individual variables on the load by means of simple longitudinal plots. Meanwhile, kernel methods have been employed with great success in the last decade. Already back in 2001, a kernel-based Support Vector Regression (SVR) approach was employed to win a competition on electricity load forecasting [7] organized by EUNITE (European Network on Intelligent Technologies for Smart Adaptive Systems). Later on, various types of kernel-based regularization methods and Support Vector Machines have been applied to predict the demand of electricity, see for instance [8], [9], [10].

Most research articles on electricity load forecasting focus on predicting a single time series representing the electricity load aggregated over a large number of nodes of the electricity network. For example, in [11], the authors investigate methods that include scenario generation for long-term load forecasting. Due to aggregation, such time series exhibit high regularity and are therefore significantly easier to forecast than load profiles at lower levels of the network. Nevertheless, making forecasts of the loads at lower levels is becoming increasingly feasible due to the availability of rich smart meter datasets, therefore the problem is attracting considerable interest in the industry.

Forecasting electricity demand at low levels of the network (such as the demand of an individual household) presents several challenges. First of all, it involves analyzing a much larger number of time series, calling for scalable techniques that can handle a very large amount of measurements. In addition, demand profiles at lower levels of the electricity network are much less regular and thus harder to predict. To tackle these challenges, recent works have investigated the use of clustering techniques for automatically aggregating multiple load time series, reporting improved predictive performance at aggregated level [12], [13], [14]. In [15], the authors investigated the use of multi-task Gaussian process models for the short-term power load forecast of a small number of cities.

In this paper, we study the problem of mid-term electricity load forecasting at the smart meter level, and we suggest to solve it by means of kernel-based multi-task learning techniques that can discover and take advantage of the relationships between multiple profiles. Kernel based multi-task learning has been studied in a variety of papers [16], [17], [18], [19] while, in recent years, the problem of learning and exploiting relationships between multiple tasks is a topic that is attracting considerable attention in the machine learning literature [20], [21], [22], [23], [24], [25], [26], [27], [28],

[29].

Herein, we develop and compare a variety of kernel-based models for medium-term electricity demand forecasting in multiple nodes, with the goal of identifying the best way to capture the complex seasonal effects that characterize such demand patterns.

A. Contributions

This paper makes 3 main contributions.

- In Section II, we formulate the problem of forecasting the demand in multiple nodes of the network as a multi-task learning problem, illustrating the usefulness of jointly learning and exploiting similarities between multiple load profiles.
- In Section III, we design kernels specifically tailored to capture the seasonal effects present in electricity load data.
- In Section IV, we expose the performance limits of the very popular additive models, showing that they are often outperformed by multiplicative kernel models. We show how recently developed multi-task learning techniques can be used to gain insights and interpretability on real demand data, while achieving state of the art predictive performance. Our experimental analysis is based on data provided by the Irish Commission for Energy Regulation (CER).

II. Electric demand forecasting as a multi-task learning problem

A. General demand forecasting model

Electric demand forecasting aims at predicting the future demand on one or multiple power lines of an electricity network. Depending on how far ahead in time the forecast is required, the corresponding estimation problem exhibits different characteristics, and influence decisions of significantly different nature. It is therefore common to classify forecasting problems in three categories: short-term forecasting (several minutes up to one week ahead), medium-term forecasting (up to 10 years ahead), and long-term forecasting (as far as several decades ahead), see [1] for a more comprehensive discussion.

Forecasting models are built starting from datasets containing one or multiple time series, each of them representing the demand measurement on a specific line in the network, ranging from highly aggregated demands in the transmission network down to the distribution network and to the demands of individual users. Missing measurements and different sampling rates contribute to make these data noisy and challenging to analyze. Moreover, defective meters at a low level in the network are hard to detect, and faulty meters can report wrong measurements before being replaced.

Being mostly driven by human activity, a variety of temporal patterns can be observed in the load data [30]. A variety of additional features can be typically extracted from the data or obtained from other sources and utilized to forecast the electricity demand. For instance, the electricity consumption is affected by weather conditions (particularly due to heating and air-conditioning), therefore variables such as temperature, humidity and irradiance are often taken into account by forecasting models. Economic indicators such as gross domestic product can be used to model trends in long-term scenarios. Finally, short-term forecasting models are typically based on time series techniques, where auto-regressive lagged values of the load itself are incorporated in the model and used to track short-term trends and deviations from stationarity.

In summary, typical forecasts of the electricity demand may depend on a variety of features that include time and calendar variables, weather and economic conditions, previously observed values of the load, and information about the node of the network where the forecast is required. A general model that takes into account the previously discussed features takes the form

$$\text{Forecast} = f\left(\underbrace{t, d, c}_{\substack{\text{Time / Calendar} \\ \text{features}}} , \underbrace{y_l, u_l}_{\substack{\text{Dynamic} \\ \text{features}}} , \underbrace{j, s_j}_{\text{Meter features}} \right), \quad (1)$$

where the dependent variables are the following:

- $t \in [0, 24)$ is the time of day expressed in hours,
- $d \in \{1, 2, \ldots, 365, 366\}$ is the day of the year,
- c is the type of day, e.g. Monday to Sunday, weekday/weekend, holiday,
- y_l is a real vector containing lagged values of the measured electric demand,
- u_l is a real vector containing measurements of lagged values of exogenous variables other than the load (such as temperature),
- j is the meter ID in the electricity network, e.g. corresponding to a specific device / customer / region,
- s_j is a vector of features describing the characteristics of the demand measured by the meter j, e.g. device / customer / region type.

B. Solving multiple demand forecasting problems by kernel-based multi-task regression

In this section, we analyze one of the many possible multi-task learning problems that naturally appear within the framework described in Section II-A, namely the problem of simultaneously predicting the demand measured at several power lines of the network. This amount to disaggregate the overall dataset over the multiple smart meters (indexed by j) and treat each one of them as a different learning task. We briefly recall the standard setup of multi-task regression and review the techniques that will be employed in Section IV to solve this forecasting problem.

In the following, we focus on multi-variate (multi-task) regression problems where the goal is to learn multiple functions $f_j : \mathcal{X} \to \mathbb{R}$ from multiple datasets of pairs $(x_{ij}, y_{ij}) \in \mathcal{X} \times \mathbb{R}$. Here, $\mathcal{X}$ is a set of input features, m denotes the number of tasks, and ℓ_j the number of examples for the j-th task (i.e. ℓ_j is the number of measurements recorded by the smart meter j). Letting $f : \mathcal{X} \times \mathbb{R}^m$ denote the vector-valued function with

components f_j, we are going to search f by minimizing the following regularization functional

$$R(f, \mathbf{L}) = \sum_{j=1}^{m} \sum_{i=1}^{\ell_j} \left(y_{ij} - f_j(x_{ij})\right)\right)^2 + \lambda \|f\|^2_{\mathcal{H}_\mathbf{L}}. \quad (2)$$

where $\lambda > 0$ is a regularization parameter, and $\mathcal{H}_\mathbf{L}$ is a Reproducing Kernel Hilbert Space (RKHS) of vector-valued functions with (matrix-valued) kernel

$$H(x_i, x_j) = K(x_i, x_j)\mathbf{L}. \quad (3)$$

Here, $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a positive semidefinite kernel called *input kernel*, and the square matrix $\mathbf{L} \in \mathbb{R}^{m \times m}$ is the *output kernel* matrix whose entries $\mathbf{L}_{jk}$ express the similarity between the tasks (output components) j and k. In view of the *representer theorem*, there exist functions $\hat{f}_j$ minimizing $R(f, \mathbf{L})$ in the form:

$$\hat{f}_j(x) = \sum_{k=1}^{m} \mathbf{L}_{jk} \sum_{i=1}^{\ell_k} c_{ik} K(x_{ik}, x). \quad (4)$$

We refer to [31] for more details about RKHS of vector-valued functions and the corresponding representer theorem.

1) Fixing $\mathbf{L} = \mathbf{I}$ (independent kernel ridge regression): Expression (4) shows that inter-task transfer is possible only when off-diagonal elements of the output kernel matrix are different from zero. Indeed, by choosing $\mathbf{L}$ equal to the identity matrix all the tasks are learned independently by solving a standard kernel regularized least squares problem

$$\hat{f}_j = \arg\min_{f_j \in \mathcal{H}} \left(\sum_{i=1}^{\ell_j} \left(y_{ij} - f_j(x_{ij})\right)^2 + \lambda \|f_j\|^2_{\mathcal{H}} \right), \quad (5)$$

where $\mathcal{H}$ is the RKHS of scalar functions with kernel K. This single-task baseline is referred to as *independent* kernel ridge regression [32].

2) Learning $\mathbf{L}$ (output kernel learning): In this section, we review a kernel-based multi-task regression approach called low-rank Output Kernel Learning (OKL), recently developed in [29]. In such approach, the functions f_j and the output kernel $\mathbf{L}$ are jointly optimized by solving the following problem

$$\min_{\mathbf{L} \in \mathbb{S}^{m,p}_+} \quad \min_{f \in \mathcal{H}_\mathbf{L}} \quad R(f, \mathbf{L}) + \lambda \mathrm{tr}(\mathbf{L}), \quad (6)$$

where $\mathbb{S}^{m,p}_+$ is the cone of positive semidefinite matrices with rank less than or equal to p. Instead of imposing a low-rank constraint or regularizing the trace of the output kernel, other type of regularizers could be tried, see e.g. [23], [33], [34]. The low-rank approach has the advantage of allowing us to tightly control the memory required to store the models.

The representer theorem (4) still applies to the inner minimization problem of (6). By plugging the expression (4) into (6), one obtains a functional that is convex quadratic with respect to both the coefficients c_{ik} and $\mathbf{L}$. Although the resulting problem is not jointly convex, the alternating minimization procedure described in [29] can be applied to obtain a minimizer. An important aspect of the method is that, by selecting the rank parameter p, is it possible to control the overall number of parameters of the model, as well as the memory requirements and the computation time to obtain a solution. More specifically, letting $\mathcal{A} = \cup_j \cup_i \{x_{ij}\}$, one can show that the solution (6) can be rewritten as

$$\hat{f}_j(x) = \sum_{k=1}^{p} b_{jk} g_k(x), \quad g_k(x) = \sum_{i=1}^{\ell} a_{ik} K(x_i, x), \quad (7)$$

where $\ell = \#\mathcal{A}$, $x_i \in \mathcal{A}$, $i = 1, \ldots, \ell$, and the coefficients b_{jk} form a low-rank factor of $\mathbf{L}$. It is therefore sufficient to store and optimize $(\ell + m)p$ parameters, which can be much smaller than $\sum_{j=1}^{m} \ell_j$.

3) Computational complexity: When performing independent kernel ridge regression ($\mathbf{L}$ is fixed as the identity matrix, see Section II-B1), solving the forecasting problems for all meters is trivially linear in the number of meters. When performing output kernel learning (Section II-B2), the complexity is also linear in the number of meters. This result is not as obvious, and we refer the reader to [29] for details.

III. Kernels for electricity load forecasting

In this section, we design kernels specifically tailored to capture the seasonal effects present in electricity load data. In order to define suitable kernels, let us have a look at electricity demand patterns. The top panel of Fig. 1 shows a typical profile for aggregated electricity load over several years (data source: French Réseau de Transport d'Electricité[1]), from which a clear yearly seasonal pattern can be observed, with higher demand in winter and lower demand in the summer. A closer look at this data also reveals typical weekly (Fig. 1, middle panel) and daily (Fig. 1, bottom panel) profiles. Correctly capturing these seasonal patterns is an crucial aspect of the problem, which can be dealt with by properly extracting and utilizing temporal and calendar features. The type of day of the week can be also taken into account: the bottom panel of Fig. 1 shows a specific week where all days have a similar profile but a difference between week days and weekend can be clearly noticed. Forecasting is particularly challenging on public holidays, and different public holidays may exhibit significantly different load profiles.

Given these observations, and using the same notations as in Section II, we introduce kernels based on the time/calendar features of model (1),

- Time-of-day kernel

$$K^t(t_1, t_2) = \exp\left(-h_T(|t_1 - t_2|)/\sigma_t\right), \quad (8)$$

- Day-of-year kernel

$$K^d(d_1, d_2) = \exp\left(-h_D(|d_1 - d_2|)/\sigma_d\right), \quad (9)$$

- Day-type kernel

$$K^c(c_1, c_2) = \begin{cases} 1 & \text{if } c_1 = c_2 \\ 0 & \text{if } c_1 \neq c_2. \end{cases}, \quad (10)$$

where $h_P(x) = \min\{x, P - x\}$ is a change of variable that yields P-periodic kernels over the square $[0, P]^2$. By observing that the Fourier transform of $\exp(-|x|)$ is non-negative, it can

[1] http://clients.rte-france.com/lang/fr/visiteurs/vie/vie_stats_conso_inst.jsp

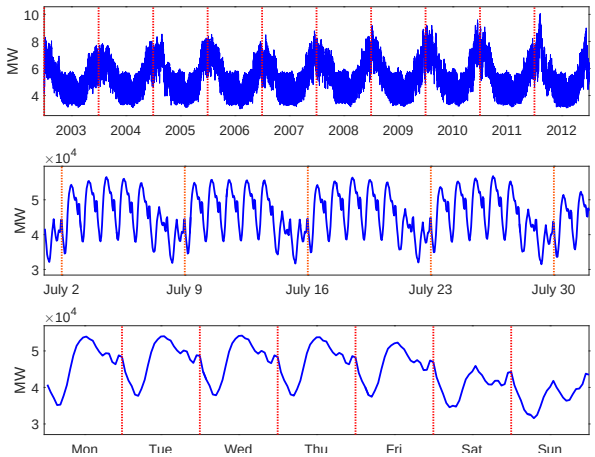


Fig. 1: Electric load data: yearly, weekly, and daily seasonal patterns can be observed in the top, middle, and bottom panels, respectively.

TABLE I: Number of meters and sparsity (percentage of missing measurements) for each customer group in the Irish CER dataset

Customer group	**Meters**	**Sparsity**
Residential	4225	0.028%
Industrial (SME)	485	0.035%
Others	1723	17%

be easily shown that periodized kernels such as K^t and K^d are positive semidefinite, see [35]. In our experiment, σ_t and σ_d were respectively set to 4 hours and 120 days. In order to define $K((t_1, d_1, c_1), (t_2, d_2, c_2))$, we combine these three kernels to define a variety of models

- Additive Models

$$K^t(t_1, t_2) + K^d(d_1, d_2)\,, \tag{11}$$
$$K^t(t_1, t_2) + K^d(d_1, d_2) + K^c(c_1, c_2)\,, \tag{12}$$

- Semi-Additive Models

$$K^d(d_1, d_2) + K^t(t_1, t_2) \cdot K^c(c_1, c_2)\,, \tag{13}$$
$$\big(K^t(t_1, t_2) + K^d(d_1, d_2)\big) \cdot K^c(c_1, c_2)\,, \tag{14}$$

- Multiplicative Models

$$K^t(t_1, t_2) \cdot K^d(d_1, d_2)\,, \tag{15}$$
$$K^t(t_1, t_2) \cdot K^d(d_1, d_2) \cdot K^c(c_1, c_2)\,, \tag{16}$$

IV. Experimental validation on smart meter data

In this section, we focus on predicting the demand of electricity measured on multiple power lines of an electricity network. Specifically, we focus on medium-term forecasts of multiple demands measured by smart meters, a multi-task learning problem where each task corresponds to one smart meter.

A. Data and pre-processing

We adopt data provided by the the Irish Commission for Energy Regulation (CER) [2], containing electric load measurements from 6435 smart meters, half-hourly sampled from July 14, 2009 to December 31, 2010 (536 days). These meters include residential customers and small-to-medium industrial sites. We consider a mid-term test scenario where the goal is to forecast the load in multiple nodes over a time horizon of 171 days, using one year of measurements to build the model. Due to the long forecasting horizon, dynamic features are not available and are therefore dropped from the general model in Eq. (1). Such model does not rely on recent measurements of the load, therefore it is able to make predictions over an arbitrarily long horizon. The load forecast for the j-th smart meter is thus simply given by $\hat{y} = f_j(t, d, c)$, where f_j are the multiple functions to be learned, taking into account time and calendar features.

From the original CER dataset, several pre-processing steps were performed. The day of the year and time of the day were extracted from the five-digit timestamps. In this dataset, the time of day is an non-zero integer indexing the number of half-hours, and therefore it should be normally in the set $\{1, 2, \ldots, 48\}$. Two meters containing time of days higher than 50 half-hours were discarded, as it was unclear how to interpret these measurements. The dataset also contains days with 46 and 50 measurements and time of days up to 50. These inconsistencies are caused by the start and the end of daylight saving time (DST) and are easily fixable. When DST starts in Ireland [3], the 1AM to 2AM hour get skipped, and half-hourly time of day indices should be $\{1, 2, 5, 6, \ldots, 48\}$. When DST starts in Ireland, the 1AM to 2AM hour "happens twice", and half-hourly time of day indices should be $\{1, 2, 3, 4, 3, 4, 5, 6, \ldots, 48\}$, instead of $\{1, 2, \ldots, 50\}$ as found in the dataset. We then downsampled each time-series from half-hourly sampling to 3-hour sampling, by averaging available measurements for each time slot of 3 hours ([12AM, 3AM), [3AM, 6AM), etc) and a total of 8 measurements per day. Our final dataset contains $m = 6433$ smart meters sampled over $\ell = 4288$ time slots. Characteristics of such pre-processed dataset are summarized in Table I.

B. Quantitative analysis

1) Learning the models: We used one year (2920 downsampled observations) for training and validation, and the remaining 1368 observations for testing. In order to perform tuning of the regularization parameter, we extracted a validation set containing a subset of the original non-test data, obtained by randomly choosing 20% time samples, equal for all the meters.

We trained independent kernel ridge regression models (see Sec. II-B1) for each measured smart meter using all the kernels from (11) to (16). We compare these models against a multi-task learning approach that simultaneously performs estimates for all the meters, and also allows us to exploit the available meter grouping information in the dataset. Specifically, we

[2] http://www.ucd.ie/issda/data/commissionforenergyregulationcer/
[3] http://www.timeanddate.com/time/change/ireland/dublin

trained two separate multi-task output kernel learning (OKL) models (see Sec. II-B2): the first is trained over all the residential meters (the union of meters labeled "residential" and "others"), and the second over industrial meters (labeled in the dataset as "SME" for "small or medium enterprise"). The maximum rank constraint for the first model was set to $p = 200$ to obtain a compact model that fits into memory, while the OKL model for SME meters was trained with full rank $p = 485$. We refer the reader to [29] for a discussion on the effect of this parameter. Both OKL models utilize the multiplicative input kernel (16), as it proves to give the highest accuracy.

2) Performance metrics: Forecasting performance can be evaluated for each time slot $i = 1, \dots, \ell$ and any arbitrary group of meters $\mathcal{G}$. For this purpose, let $\mathcal{G}_i$ denote the subset of $\mathcal{G}$ for which measurements are available in the i-th time slot. We define two different metrics in order to evaluate the accuracy of the aggregate forecast and the accuracy of the individual forecasts. Let us define

$$\mathrm{APE}(i, \mathcal{G}) = 100 \left| \frac{\sum_{j \in \mathcal{G}_i} y_{ij} - \sum_{j \in \mathcal{G}_i} f_j(t_i, d_i, c_i)}{\sum_{j \in \mathcal{G}_i} y_{ij}} \right| , \quad (17)$$

$$\mathrm{NAE}(i, \mathcal{G}) = \frac{\sum_{j \in \mathcal{G}_i} |y_{ij} - f_j(t_i, d_i, c_i)|}{\sum_{j \in \mathcal{G}_i} y_{ij}} . \quad (18)$$

$\mathrm{APE}(i, \mathcal{G})$ measures the absolute percentage error incurred at time i when forecasting the aggregated demand using the sum of the forecasts in $\mathcal{G}$. On the other hand, $\mathrm{NAE}(i, \mathcal{G})$ is the sum of the forecasting errors over individual tasks, relative to the naive baseline of predicting $f_j(t_i, d_i, c_i) = 0$ for all i, j. Since the demand values y_{ij} are always non-negative, the two metrics are undefined only for those groups on meters for which the cumulative demand in the i-th time slot is identically zero, or no measurements are available for any of the meters. We compute the average and standard deviation of these two metrics over all the observations in the test period. In particular, we define the (aggregated) mean absolute percentage error (MAPE) and the mean normalized absolute error (MNAE)

$$\mathrm{MAPE}(\mathcal{G}) = \frac{1}{\# \, T} \sum_{i \in T} \mathrm{APE}(i, \mathcal{G}) , \quad (19)$$

$$\mathrm{MNAE}(\mathcal{G}) = \frac{1}{\# \, T} \sum_{i \in T} \mathrm{NAE}(i, \mathcal{G}) . \quad (20)$$

Noting the aggregate signal $z_i = \sum_j y_{ij}$ and forecast of aggregate signal $\hat{z}_i = \sum_j f_j(t_i, d_i, c_i)$, one can obtain $\mathrm{MAPE} = 100 \, \frac{1}{\# \, T} \sum_{i \in T} |\frac{z_i - \hat{z}_i}{z_i}|$. In other words, the MAPE defined from Eq. (17) and (19) is the standard MAPE definition for the aggregate of meters in the group $\mathcal{G}$.

We also compute the standard errors on the MAPE and MNAE as the empirical standard deviations of, respectively, APE and NAE, divided by the number of observations. Finally, we report the p-values of the Welch t-test, computed from the average and standard deviation of the previously defined metrics, and the number of smart meters.

3) Results: Fig. 2 illustrates the challenges of medium-term forecasting at low network level versus forecasting aggregated demands. We analyze the measured load and the corresponding forecast over a window of 5 weeks within the test period. In the top panel, one can see the aggregated load and the corresponding forecast obtained as the sum of all disaggregated forecasts obtained using model (16). The kernel-based forecasts are rather accurate overall, only slightly estimating the total load during the Christmas week, a particularly problematic period to predict. In the middle panel, the measured load for a single SME meter is compared with the corresponding forecast. The varying demand profiles of different days of the week are captured rather well by the model. Again, there is a larger error over the Christmas week, caused by a sudden drop of the demand to a low baseline value (probably due to interruption of business activities followed by a slow resumption in the subsequent days). This leads the models to over-estimate the load, though the model learned with a multi-task approach is less affected. Finally, the bottom panel shows the electricity demand of a residential customer, characterized by rapid variations with sharp consumption peaks and irregular patterns, that make the forecast even more difficult.

Fig. 3 reports the performance of all methods over the full set of 6433 smart meters, as well as disaggregated performance measures over each group from Table I. We start by analyzing the performance of the additive models, which are probably the most widely adopted in the literature. By comparing the performance of models (11) and (12), we can observe that adding a constant bias specific to the type of the day of week (kernel K^c) does not necessarily improve the accuracy of the model. The overall MNAE and MAPE are in fact higher for model (12). Semi-additive models where the type of day of the week is utilized to switch between different profiles yields a significant improvement in performance. The two semi-additive models (13) and (14) achieve similar performance over the groups residential and others. However, for the SME customer group, model (14) is better in terms of both MNAE and MAPE. In previous works such as [5], semi-additive models of the form (13) have been proposed to switch between different daily patterns, depending on the type of day. Interestingly, our results show that in certain situations, such as when modeling industrial customers, it is even better to switch the overall sum of the daily pattern and the yearly pattern. We took a step even further by utilizing fully multiplicative models (15) and (16). The multiplicative model (15) pools over different days of the week, while (16) learns independent models for each day. While the former is not always better than the semi-additive models, the latter significantly outperforms them. We can conclude that a multiplicative kernel structure (16) is the best at forecasting time series with yearly, weekly and daily seasonal effects, both overall and for each customer group. Such conclusion is aligned with recent results presented in [36], where tensor product basis functions were utilized to capture weekly and yearly seasonalities in the simpler context of load forecasting for a single highly aggregated time series. A further performance improvement can be obtained by utilizing a multi-task learning approach, where correlation between electricity demand behavior of multiple customers

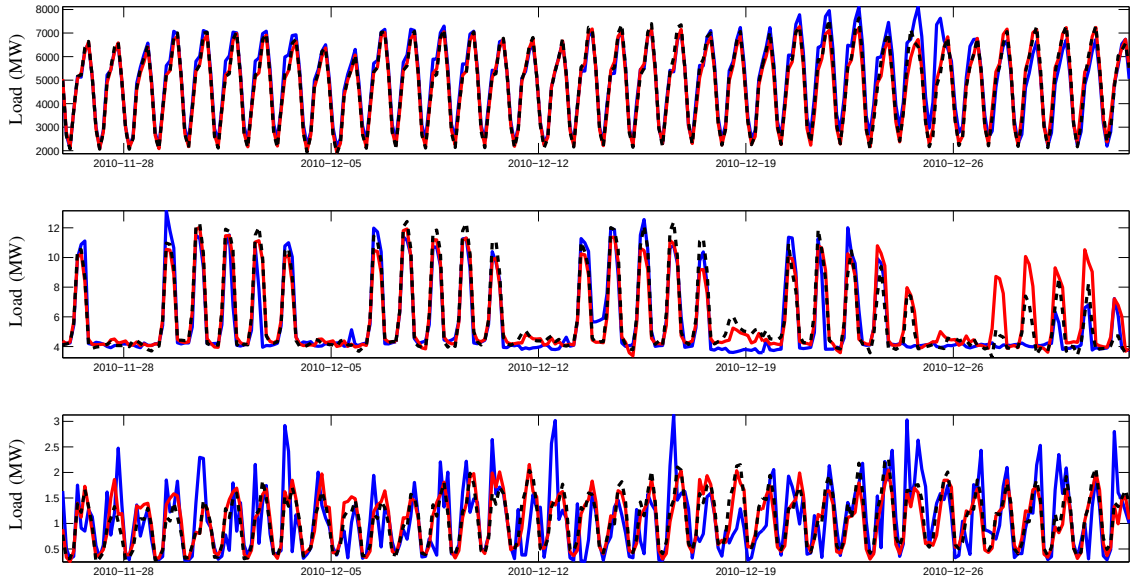


Fig. 2: CER data: measured load (blue curves) and corresponding forecast with independent (red curve) and multi-task (black dashed curve) for the aggregated demand (top panel), a single SME meter (mid panel), and a single residential meter (bottom panel). Measurements are shown over 5 weeks of the test period. All forecasts are obtained using a multiplicative kernel model (16).

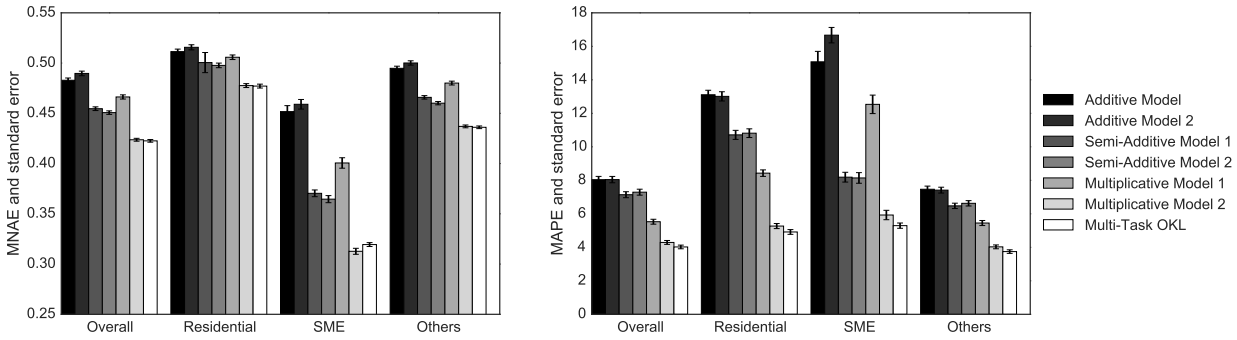
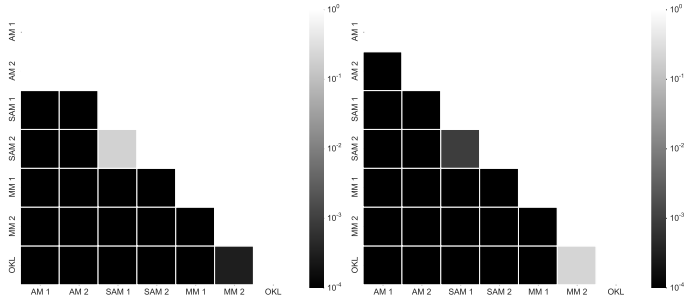


Fig. 3: Accuracy of the different medium-term forecasting methods on the CER dataset

is learned and exploited. As Fig. 3 shows, the multi-task OKL approach provides the lowest MNAE over all the meters performance, residential and others, and second lowest MNAE for SME (only 2% higher than the lowest for this group). The multi-task learning approach also provides the lowest mean aggregated MAPE, overall and for each customer groups. Finally, the multi-task approach is more robust, as the temporal standard deviation is the lowest for both NAE and APE. Again, such robustness can be observed overall the customers as well as for each customer group. In particular, it is worth mentioning a 44% improvement of the standard deviation of the APE for SME meters, compared to the best single task model that uses the multiplicative model (16).

Fig. 4 presents the p-values of the Welch t-test, computed for all pairs of models, for both the overall NAE and APE. Using a threshold of $\alpha = 10^{-2}$, most models are statistically different from one to each other. In particular, we notice that using any method that uses a multiplicative kernel (Multiplicative Model 1, Multiplicative Model 2 and Multi-Task OKL) are



(a) NAE (b) APE

Fig. 4: p-values of Welch t-test between the overall accuracies of all methods on the CER dataset

statistically superior to any listed method that uses an additive or semi-additive kernel. Furthermore, the multi-Task OKL is statistically better than all other methods in terms of MNAE,

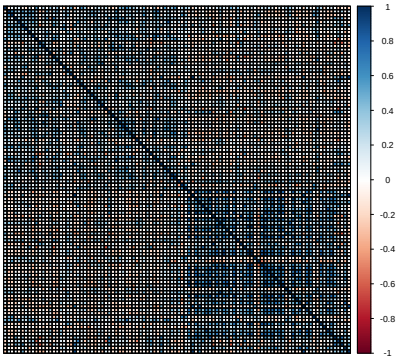


Fig. 5: CER data: entries of the normalized output kernel $\mathbf{L}_n$ for a subset containing 50 'residential' and 50 'SME (small or medium enterprise)' customers.

and than all methods except Multiplicative Model 2 in terms of MAPE.

In addition to improving forecasting accuracy, the low-rank multi-task learning model is significantly more compact in terms of number of parameters. For all the single-task methods (with additive, semi-additive and multiplicative kernels), the number of parameters is equal to the overall number of training observations $\sum_{j=1}^{m} \ell_j$. In our experiment, this amounts to about 13 million parameters (precisely 12785524 parameters). The low-rank output kernel learning method models each prediction function $\hat{f}_j$ as a linear combination of p latent functions, shared by all tasks (see Sec. II-B2). These p functions g_k can be seen as typical load profiles. As a consequence, only $(\ell + m)p$ parameters are required to learn the prediction functions for all smart meters. In our experiment, this gave a total of about 3 million parameters (precisely 3016310) thus producing a model that is about 4.24 times more compact, in addition to being more accurate.

C. Smart Meter Load Profile Analysis via Multi-Task OKL

In this Section, we evaluate additional benefits of using the multi-task OKL approach for electric load forecasting problems. For this section, we built a low-rank OKL model using the data from all smart meters.

In order to visualize the relationships between several smart meters, we define the normalized output kernel $\mathbf{L}_n \in \mathbb{R}^{m \times m}$ as

$$(\mathbf{L}_n)_{ij} = \frac{\mathbf{L}_{ij}}{\sqrt{\mathbf{L}_{ii} \times \mathbf{L}_{jj}}}, \qquad i, j = 1, \ldots, m. \qquad (21)$$

Figure 5 shows the entries of the learnt normalized output kernel $\mathbf{L}_n$ corresponding to a subset of nodes including 50 residential and 50 'SME (small or medium enterprise)' meters, where a correlation structure consistent with the labeling can be observed.

A further possibility offered by the low-rank model is to obtain a small set of p functions that generate all the estimated tasks by linear combination, namely the functions g_k $(k = 1, \ldots, p)$ in expression (7). In our scenario, these functions are interpretable as 'typical load profiles'. Figure 6 shows few of them ($p = 10$) over the horizon of approximately one month, revealing a variety of typical weekly and daily load patterns.

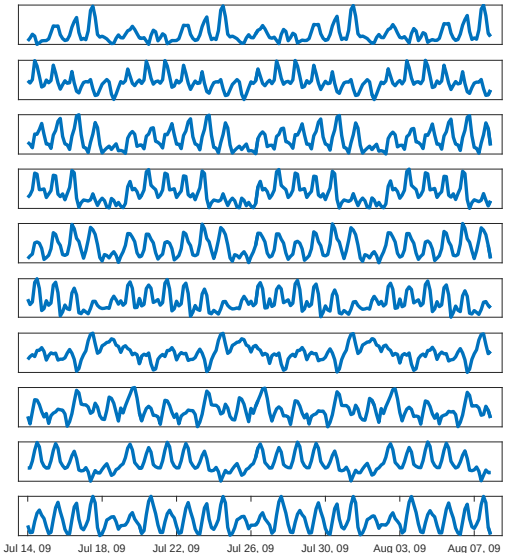


Fig. 6: CER Data: Typical load profiles displayed over the horizon of one month, obtained from a low-rank OKL model with $p = 10$.

V. Discussion and Conclusions

Our analysis shows that kernel-based multi-task learning is effective for the resolution of electric load forecasting problems. Focusing on the challenging problem of forecasting the electric load of individual customers, we designed kernels that take into account relevant multiple seasonality patterns. We demonstrated the clear benefits of multiplicative kernel models over additive or semi-additive models. Our results suggest a new modeling direction, as opposed to the (generalized) additive models, widely employed in the energy community. We illustrated further performance gain made possible by using a multi-task learning approach over a large number of single-tasks baselines. While recent studies reported MAPE around 3% for the short-term forecasting of an aggregated signal of a few thousands of smart meters e.g. [13], our method achieves a MAPE of 4% on a medium term forecasting scenario, which is a much harder problem as neither auto-regressive terms nor accurate weather forecasts are available.

The ideas and results presented in this paper open a wide range of considerations. First of all, they suggest that electricity demand data can be used as natural test benchmarks for multi-task learning methods. In addition, these problems motivate developing new techniques that allow to incorporate more complex task relationships structures taking into account, for instance, topological and physical constraints from the electricity network. The development of online methods that can automatically discover relationships between multiple tasks seems to be particularly important for short-term load forecasting scenarios. Finally, combining online multi-task learning methods with topological network constraints would allow to start tackling very complex scenarios such as forecasting on a full electricity network with dynamic reconfigurations.

REFERENCES

[1] S. Soliman and A. Al-Kandari, *Electrical load forecasting: modeling and model construction*. Elsevier, 2010.

[2] H. S. Hippert, C. E. Pedreira, and R. C. Souza, "Neural networks for short-term load forecasting: A review and evaluation," *Power Systems, IEEE Transactions on*, vol. 16, no. 1, pp. 44–55, 2001.

[3] T. Hastie and R. Tibshirani, *Generalized additive models*. CRC Press, 1990, vol. 43.

[4] S. Fan and R. Hyndman, "Short-term load forecasting based on a semi-parametric additive model," *Power Systems, IEEE Transactions on*, vol. 27, no. 1, pp. 134–141, 2012.

[5] A. Ba, M. Sinn, Y. Goude, and P. Pompey, "Adaptive learning of smoothing functions: application to electricity load forecasting," in *Advances in Neural Information Processing Systems 25 (NIPS 2012)*, 2012, pp. 2519–2527.

[6] R. Nedellec, J. Cugliari, and Y. Goude, "Gefcom2012: Electric load forecasting and backcasting with semi-parametric models," *International Journal of Forecasting*, 2013.

[7] B. Chen, M. Chang, and C. Lin, "Load forecasting using support vector machines: A study on EUNITE competition 2001," *Power Systems, IEEE Transactions on*, vol. 19, no. 4, pp. 1821–1830, 2004.

[8] M. Espinoza, J. Suykens, R. Belmans, and B. De Moor, "Electric load forecasting," *Control Systems, IEEE*, vol. 27, no. 5, pp. 43–57, 2007.

[9] W.-C. Hong, "Electric load forecasting by support vector model," *Applied Mathematical Modelling*, vol. 33, no. 5, pp. 2444–2454, 2009.

[10] E. Elattar, J. Goulermas, and H. Wu, "Electric load forecasting based on locally weighted support vector regression," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 40, no. 4, pp. 438–447, 2010.

[11] T. Hong, J. Wilson, and J. Xie, "Long term probabilistic load forecasting and normalization with hourly information," *Smart Grid, IEEE Transactions on*, vol. 5, no. 1, pp. 456–462, 2014.

[12] C. Alzate, M. Espinoza, B. De Moor, and J. A. Suykens, "Identifying customer profiles in power load time series using spectral clustering," in *Artificial Neural Networks–ICANN 2009*. Springer, 2009, pp. 315–324.

[13] C. Alzate and M. Sinn, "Improved electricity load forecasting via kernel spectral clustering of smart meters," in *2013 IEEE 13th International Conference on Data Mining (ICDM)*. IEEE, 2013, pp. 943–948.

[14] S. Humeau, T. K. Wijaya, M. Vasirani, and K. Aberer, "Electricity load forecasting for residential customers: Exploiting aggregation and correlation between households," in *Sustainable Internet and ICT for Sustainability (SustainIT), 2013*. IEEE, 2013, pp. 1–6.

[15] Y. Zhang, G. Luo, and F. Pu, "Power load forecasting based on multi-task gaussian process," in *Proceedings of the IFAC 19th World Congress*, 2014, pp. 3651–3656.

[16] C. Micchelli and M. Pontil, "Kernels for multi–task learning." in *Advances in Neural Information Processing Systems 17 (NIPS 2004)*, 2004.

[17] T. Evgeniou, C. A. Micchelli, and M. Pontil, "Learning multiple tasks with kernel methods," *Journal of Machine Learning Research*, vol. 6, pp. 615–637, 2005.

[18] A. Argyriou, T. Evgeniou, and M. Pontil, "Multi-task feature learning," in *Advances in Neural Information Processing Systems 19 (NIPS 2006)*, vol. 19. MIT; 1998, 2006, p. 41.

[19] E. Bonilla, K. Chai, and C. Williams, "Multi-task Gaussian process prediction." in *Advances in Neural Information Processing Systems 20 (NIPS 2007)*, vol. 20, 2007, pp. 153–160.

[20] L. Jacob, F. Bach, and J. Vert, "Clustered multi-task learning: A convex formulation," in *Advances in Neural Information Processing Systems 21 (NIPS 2008)*, vol. 21, 2008, pp. 745–752.

[21] J. Zhang, Z. Ghahramani, and Y. Yang, "Flexible latent variable models for multi-task learning," *Machine Learning*, vol. 73, no. 3, pp. 221–242, 2008.

[22] Y. Zhang and D.-Y. Yeung, "A convex formulation for learning task relationships in multi-task learning," in *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence (UAI)*, Catalina Island, CA, USA, 2010, pp. 733–442.

[23] F. Dinuzzo, C. S. Ong, P. Gehler, and G. Pillonetto, "Learning output kernels with block coordinate descent," in *Proceedings of the 28th Annual International Conference on Machine Learning*, Bellevue, WA, USA, 2011.

[24] C. Archambeau, S. Guo, and O. Zoeter, "Sparse bayesian multi-task learning." in *Advances in Neural Information Processing Systems 24 (NIPS 2011)*, vol. 1, 2011, p. 41.

[25] Z. Kang, K. Grauman, and F. Sha, "Learning with whom to share in multi-task feature learning," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 521–528.

[26] A. Saha, P. Rai, H. Daumé III, and S. Venkatasubramanian, "Online learning of multiple tasks and their relationships," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, Ft. Lauderdale, Florida, 2011.

[27] A. Kumar and H. Daumé III, *Learning task grouping and overlap in multi-task learning*. ICML, 2012.

[28] B. Romera-Paredes, A. Argyriou, N. Berthouze, and M. Pontil, "Exploiting unrelated tasks in multi-task learning." *Journal of Machine Learning Research - Proceedings Track*, vol. 22, pp. 951–959, 2012.

[29] F. Dinuzzo, "Learning output kernels for multi-task problems," *Neurocomputing*, vol. 118, pp. 119–126, 2013.

[30] T. Hong, "Short term electric load forecasting," Ph.D. dissertation, 2010.

[31] C. A. Micchelli and M. Pontil, "On learning vector-valued functions," *Neural Computation*, vol. 17, pp. 177–204, 2005.

[32] K. P. Murphy, *Machine Learning: a probabilistic perspective*. The MIT press, 2012, ch. 14.4.3, pp. 492–493.

[33] F. Dinuzzo, C. Ong, and K. Fukumizu, "Output kernel learning methods," in *Regularization, Optimization, Kernels, and Support Vector Machines,*, M. A. A. Suykens, J; Signoretto, Ed. CRC Press, 2014.

[34] C. Ciliberto, T. Poggio, and L. Rosasco, "Convex learning of multiple tasks and their structure," in *International Conference on Machine Learning*, 2015.

[35] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, ser. (Adaptive Computation and Machine Learning). MIT Press, 2001.

[36] A. Guerini and G. De Nicolao, "Long-term electric load forecasting: A torus-based approach," in *Proceedings of the European Control Conference*. IEEE, 2015.

Jean-Baptiste Fiot is a Research Scientist at IBM Research - Ireland since December 2013. He received a Ph.D. degree in Applied Mathematics in 2013 from Paris Dauphine University in France, a Master degree in Applied Mathematics in 2009 from Ecole Nationale Superieure de Cachan in France, and a Master degree in Engineering in 2009 from Ecole Centrale Paris in France. Before joining IBM, he held Research positions in Paris Dauphine University in France, in Samsung Advanced Institute of Technology (SAIT) in South Korea, and in CSIRO - Australian e-Health Research Centre (AeHRC) in Australia. He was awarded the Best Student Paper Award in the VIPIMAGE 2011 conference, and the Thesis Prize 2014 of the Dauphine Foundation. His research interests include machine learning, signal and image processing, and optimization.

Francesco Dinuzzo is a Research Scientist at Amazon.com Seattle since September 2015. Prior to that, he has conducted research at IBM Research Dublin (Ireland), and at the Max Planck Institute for Intelligent Systems, Tbingen (Germany). He received a Ph.D. degree in Mathematics and Statistics in 2011, and a Master degree in Computer Science Engineering in 2007, both from the University of Pavia (Italy). He also received a Master degree in Science and Technology in 2007 from Istituto Universitario di Studi Superiori (Italy). He has held visiting positions at Massachusetts Institute of Technology (Center for Biological and Computational Learning), National Taiwan University (Machine Learning Group), ETH Zrich (Machine Learning Laboratory), and Institute of Statistical Mathematics, Tokyo. His research interests include machine learning, forecasting, optimization, non-linear control, system identification, regularization methods, and distributed computation.