Average Residential Electricity Consumption Prediction with Mean-Regularized Multi-Task Learning

Mahrokh Ghoddousi Boroujeni (317337)[1]

I. Abstract

In this project, we develop a learning algorithm for average residential electricity consumption prediction in a smart grid. To this end, we utilize tools from Federated Learning (FL), Multi-Task Learning (MTL), and distributed optimization. The predictions are close to real values, witnessing the high accuracy of the proposed solution.

II. Introduction

Smart grids bring about automation in managing the electricity networks by establishing a two-way interactive system through fetching information from the household and supply ends in discrete time intervals. This fine-grained data enables households to adjust their energy exchange with the grid accordingly and improves the overall reliability and efficiency of the transmission lines. While smart residential electricity management sounds a promising approach for balancing the energy demand, it requires the participating households to predict the energy consumed by the whole network ahead of time.

Even in the presence of a trusted medium for measuring the total consumption of all households in each time interval, this value cannot be given to households as it might leak information about other participants, such as their consumption values. We provide an algorithm for predicting the average electricity consumption of all houses in short intervals by each household while preserving the privacy of individual houses. The proposed algorithm only uses previous measurements in the household level, that are available locally, and weather and calendar information, that are public knowledge, and therefore, does not pose privacy risks.

III. Problem Statement

In this section, we formulate average consumption prediction as a regression problem and introduce the optimization problem for finding the model parameters.

a) Problem Setup: Consider a smart grid consisted of $N \in \mathbf{N}$ households, $\mathcal{H}_1, \cdots, \mathcal{H}_N$, and an operator to coordinate the information flow. Each household is equipped with a smart meter that records electricity consumption in 30_{min} intervals. Let $y_n(t)$ and $\overline{y}(t)$ denote the consumption of $\mathcal{H}_n$ and the average consumption of all households at interval t, i.e. $\overline{y}(t) := \frac{1}{N}\sum_{n=1}^{N} y_n(t)$.

mahrokh.ghoddousiboroujeni@epfl.ch

b) Goal: As introduced in Sec. II, the goal is to train N models, one per household, to predict $\overline{y}(t{+}1)$ at time t. The features used by $\mathcal{H}_n$, denoted by $\mathbf{x}_n(t) \in \mathbf{R}^d$, are historical measurements by $\mathcal{H}_n$, $y_n(t-t_1), \cdots, y_n(t-t_{d'})$, and weather and calendar information of $t{+}1$. Previous pairs of the features and target variable form the training dataset for $\mathcal{H}_n$, $\mathcal{D}_n = (\mathbf{y}_n, X_n) = \{(y_n(t{+}1), \mathbf{x}_n(t)) | \forall t \textit{ measured}\}$. We emphasize again that $\mathcal{H}_n$ can only use the locally available data for privacy reasons, therefore, no $y_{n'}(t)$ and $\overline{y}(t)$ terms appear in $\mathcal{D}_n$ for any $n' \neq n$. Feature selection procedure is further detailed in Sec. V-A.0.c.

c) Challenges: Using a linear model with parameters $\mathbf{w}_n^* \in \mathbf{R}^d$ for $\mathcal{H}_n$ obtains $\overline{y}(t{+}1) \approx \mathbf{x}_n^T(t)\mathbf{w}_n^*$. Conventional regression algorithms require $\mathcal{H}_n$ to solve

$$\min_{\mathbf{w}_n} \sum_{\mathcal{D}_n} \ell(\overline{y}(t{+}1), \mathbf{x}_n^T(t)\mathbf{w}_n) + \mathcal{R}(\mathbf{w}_n), \qquad (1)$$

where ℓ and $\mathcal{R}$ are prediction error and weight penalty functions respectively. The objective per. 1 depends on the average consumption, $\overline{y}(t{+}1)$, which is not known by $\mathcal{H}_n$ and hence, cannot he evaluated. Therefore, naive formulation of the average consumption prediction problem fails as it leads to optimizing an unknown function. We propose an algorithm that approximates household-level models by leveraging information from training signals of other households, while retaining the privacy of local datasets.

d) Literature Review: Electricity consumption prediction has been an active field of study, where researchers either focus on predicting the total/average consumption of a group of consumers or a single building [5]. To the best of our knowledge, no studies have focused on predicting the average consumption at building level, despite its' arising pivotal role.

IV. Proposed Algorithm

As discussed in III-.0.c, the households cannot find the optimal parameters for a model that predicts the average consumption from their local data. We develop two methods for incorporating consumption patterns of all households into local household-level models. Both methods assume the existence of a server to coordinate the training and minimize MSE as prediction loss.

A. Two-Stage Inference

A natural option for learning the average consumption pattern of households is to first fit personal models for

each household to predict its' own consumption and then average the parameters,

$$\begin{cases} \mathbf{w}_n^* = \arg\min_{\mathbf{w}_n} \|\mathbf{y}_n - X_n \mathbf{w}_n\|^2 + \lambda_w \|\mathbf{w}_n\|_2^2 \\ \mathbf{w}_n^{ts} = \mathbf{w}^{ts} = \sum_{n=1}^{N} |\mathcal{D}_n| \, \mathbf{w}_n^* \, / \sum_{n=1}^{N} |\mathcal{D}_n| \end{cases}$$

In the above, $\mathbf{w}_n^* \in \mathbf{R}^d$ is the optimal regularized parameters for $\mathcal{H}_n$ to predict its' own consumption, y_n, and $\mathbf{w}^{ts} \in \mathbf{R}^d$ is the two-stage inference (TSI) solution which is the same for all households.

Procedure: Calculating $\mathbf{w}^{ts}$ is straight forward: 1) the server sends training instructions to households, such as the optimization method, the learning rate, ... 2) the households follow the instructions to train their personal model and send back the parameters $\mathbf{w}_n^*$ and the number of training points they used, $|\mathcal{D}_n|$, to the server 3) the server calculates and publishes $\mathbf{w}^{ts}$.

Properties: The two-stage algorithm is a simple method that obtains the same model for all households. Information transfer among households occurs by communicating model parameters and does not require direct access to other households datasets, in line with the principal problem outlines. However, sharing the results only upon training completion is a limiting assumption and leaves space for further improvement.

B. Mean-Regularized Multi-Task Learning

The second approach, fits separate models for individuals and steers the parameters towards a shared model through a penalization term. This scheme can be described by the Mean-Regularized Multi-Task Learning framework [1], where fitting models for different households are referred to as tasks and mean regularization indicates that the models are close to a shared model. Assuming that these N tasks are *similar*, the objective to be minimized is:

$$\mathbf{w}^*, \mathbf{w}_0^* = \arg\min_{\mathbf{w}, \mathbf{w}_0} \sum_{n=1}^{N} \|\mathbf{y}_n - \mathbf{w}_n^T X_n\|_2^2 + \lambda_w \|\mathbf{w}_n\|_2^2 + \lambda \|\mathbf{w}_n - \mathbf{w}_0\|_2^2, \tag{2}$$

where $\mathbf{w} = [\mathbf{w}_1^T, \cdots, \mathbf{w}_N^T]^T$. The objective in (2) regularizes the model to have small coefficients, through λ_w, and be close to each other, through λ.

Derivating (2) with respect to $\mathbf{w}_0$ shows that the optimal shared model is in fact the average of all optimal local models, $\mathbf{w}_0^* = 1/N \sum_{n=1}^{N} \mathbf{w}_n^*$. This allows us to drop $\mathbf{w}_0$ from optimization variables in (2). We then estimate all $\mathbf{w}_n^*$ *simultaneously* and compute $\mathbf{w}_0^*$.

Procedure: The mean model, $\mathbf{w}_0$, ties the optimization tasks of all households together and avoids solving Eq. (2) in a distributed fashion, that is necessary to preserve the privacy of local datasets, $\mathcal{D}_n$. A common approach [3], [6] is to introduce dual variables, one per training sample, and solve the conjugate dual problem. However, the resulting problem is extremely high-dimensional and thus, not applicable to our application domain. We employ an iterative method that fixes $\mathbf{w}_0$ to optimize over $\mathbf{w}_n$ locally, computes the new average model, and iterates until convergence. To the best of our knowledge, no MTL method takes a similar approach for decoupling the optimization problems.

Algorithm 1 Recursive Mean-Reg MTL

procedure MEANREGMTL(*households*)
  operator sends training instructions
  households initialize local models $\mathbf{w}_n^{(0)}$
  for $i = 0; i <=$ *max outer iters*$; i++$ **do**
    $\mathbf{w}_0^{(i+1)} \leftarrow \frac{1}{N} \sum_{n=1}^{N} \mathbf{w}_n^{(i)}$
    for all households (in parallel) **do**
      set $\mathbf{w}_0 = \mathbf{w}_0^{(i+1)}$ in (2)
      run *inner iters* of Adam for (2)
      to find $\mathbf{w}_n^{(i+1)}$
      send the server $\mathbf{w}_n^{(i+1)}$
    end for
  end for
  return shared model $\mathbf{w}_0^*$, local models $\mathbf{w}_n^*$
end procedure

Properties: The MTL algorithm diffuses weights among households in every iteration of the outer loop and hence, exploits communication advantages more than the TSI method. A main difference of the consumption prediction sketch from the general MTL scenario is that the data is distributed among different households, i.e. $\mathcal{D}_n$ is only accessible by $\mathcal{H}_n$, which is well-handled by Alg. 1. The method returns $\mathbf{w}_n^*$ and $\mathbf{w}_0^*$, both of which can be used for prediction by $\mathcal{H}_n$. The first model is more inclined towards capturing personal differences, while the later mixes all households without prioritizing the model user, $\mathcal{H}_n$. A performance comparison between the two models, $\mathbf{w}_n^*$ and $\mathbf{w}_0^*$, cannot be explicitly asserted and might be different among households. Note that w_n and w_0 become more similar as λ increases because being similar is encouraged more per (2).

V. SIMULATIONS AND RESULTS

A. Dataset and Features

In this part, we introduce the dataset used in experiments and discuss feature extraction and selection methods for each household.

a) Data Description: The proposed methods are evaluated on an open-source dataset of smart grids in London[2]. Three types of data were recorded between November 2011 and February 2014: (i) electricity consumption of 5,567 households in half-hour resolution,

(ii) weather measurements for London area in one-hour resolution, (iii) calendar information such as date, time of day, and public holidays, (iv) social and financial information about households(refer to [2] for details).
b) Pre-processing: The pre-processing phase includes extracting useful features from the datasets, cleaning and removing missing values, and min-max scaling. All steps are detailed in *part 1* of *Visualization.ipynb*.
c) Exploratory Data Analysis: An extensive exploratory data analysis was performed to familiarize with the nature of the consumption time-series and identify important features in *part 2* of *Visualization.ipynb*. The most significant findings are: (i) some weather and calendar features were found perpendicular to consumption, (ii) for each household, the consumption patterns in a fixed day of the week are similar (e.g. Mondays look alike), (iii) public holidays extensively affect consumption. These observations allow us to drop some features from the study and motivates us to train distinct models for each day of the week. In the rest of this report, we focus on consumption prediction on Thursdays which were not a holiday.
d) Auto-regressors: Auto-regressive (AR) features, i.e. past recent consumptions of an specific household, are prominently engaged in consumption prediction literature, motivated by the slow change in people behaviour [4]. Including more AR features maximizes the explanatory power of our model, but could increase the variance in prediction and cause overfitting.
e) Feature Selection: To reduce the number of features, we test two feature selection methods: (i) Partial auto-correlation function (PACF) plot shows the amount of auto-correlation at a certain lag that is not explained by lower-order auto-correlations. The peaks in this plot contain information and are candidate features (c.f. *part 3* of *Visualization.ipynb*). (ii) By running recursive feature elimination on a linear model that predicts the consumption of a single household from its' own data. The number of selected features is tuned by cross-validation and adjusted R^2 score is used for model evaluation (c.f. *LinReg.ipynb*). According to the results of both methods, which mostly overlap, we selected the same hour of day up to two time steps before in the past 7 days (lags $48 * k + i$ for $0 \leq k \leq 7$, $0 \leq i \leq 2$ except $k = i = 0$) These features are also justified by the daily-periodic nature of households' habits.

B. Results

This section includes the results of predicting the average consumption of all households with the TSI and MTL models. To have a ground truth, we assume the average consumptions are available and train a model for predicting them directly, as per objective (1). Note that training such model is not possible in a realistic setup and is submitted for benchmarking.

The weight penalty, λ_w, had no effect in our experiments and we let $\lambda_w = 0$. The MTL algorithm uses λ to adjust the penalty term, which we tune by grid searching. All methods take a few iterations to converge and are fast. For the optimization part, we tested Adam, SGD, and the explicit MSE solutions and found Adam the best optimizer. We run the methods for 10 randomly selected households, calculate train and test MSE for each of them, and plot the average MSE over all households in Fig. 1. For MTL, we also illustrate the 95% confidence interval for MSE of households.

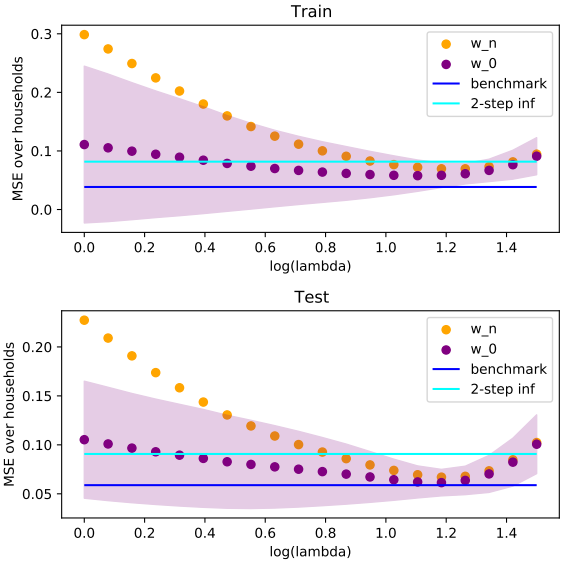

Fig. 1: MSE of different methods on train and test data

For small λ, the MTL solution is not good but as λ grows, information engagement becomes stronger and MTL outperforms TSI. Surprisingly, $\mathbf{w}_0^*$ reaches an average test MSE of 0.061 at the lowest that is so close to the benchmark, 0.059. The narrow confidence bound near the optimal λ, means that performance was good for all households. For large λ, the performance deteriorates due to strong penalization.

Next we compare the two MTL models, $\mathbf{w}_0^*$ and $\mathbf{w}_n^*$. For 3 households, $\mathbf{w}_n^*$ obtains more accurate predictions but in average, $\mathbf{w}_0^*$ is better for all λ. Additionally, $\mathbf{w}_n^*$ yields wide confidence bounds that make it inferior. This is an important issue since model performance cannot be evaluated by households and the selected models must have narrow confidence bounds.

VI. Discussions

In this project, we studied average electricity consumption prediction at household-level that was not found in the literature. We propose two methods, a naive one and a more advanced one, for this task, both of which preserve the locality of households datasets and communicate model parameters among them. The

solution based on Multi-Task Learning obtains results that are very close to the impractical benchmark results. Our algorithms do not put strict assumptions or require complicated infrastructure.

REFERENCES

[1] R Caruana. "Multitask Learning". In: *Machine Learning* 28 (1997).

[2] Jean-Michel D. *Smart meters in London*. Feb. 2019. URL: https://www.kaggle.com/jeanmidev/smart-meters-in-london.

[3] Theodoros Evgeniou and Massimiliano Pontil. "Regularized Multi–Task Learning". In: 2004.

[4] Bishnu Nepal et al. "Electricity load forecasting using clustering and ARIMA model for energy management in buildings". In: 2020.

[5] S. Seyedzadeh, F. Rahimian and I. Glesk. "Machine learning for estimation of building energy consumption and performance: a review". In: *Vis. in Eng.* 6 (2018).

[6] Virginia Smith et al. "Federated Multi-Task Learning". In: *CoRR* abs/1705.10467 (2017).