Autoregressive Model for Individual Consumption Data - Sparsity Recovery and Significance Test

Pan Li and Baosen Zhang
Electrical Engineering
University of Washington
Seattle, WA 98195
Email: {pli69, zhangbao}@uw.edu

Yang Weng and Ram Rajagopal
Civil and Environmental Engineering
Stanford University
Stanford, USA 94035
Email: {yangweng, ramr}@stanford.edu

Abstract—Understanding consumer flexibility and behavior patterns is becoming increasingly vital to the design of robust and efficient energy saving programs. Accurate prediction of consumption is a key part to this understanding. Existing prediction methods usually have high relative errors that can be larger than 30%. In this paper, we explore sparsity in users’ past data and relationship between different users to increase prediction accuracy. We show that using LASSO and Granger Causality techniques, prediction accuracy can be significantly improved in comparison to existing algorithms. We use mean absolute percentage error (MAPE) as the criteria.

Index Terms—Load forecasting, prediction of consumption, sparsity, LASSO, granger causality.

I. INTRODUCTION

A. Background

Environmental concerns and the economics of efficiently using energy resources have motivated nations, governments and technology providers to directly engage with individual customers. With recent technological advances, smart meters capable of bidirectional communication and remote control have been brought into households. In particular, it measures consumption data at much higher resolution, i.e., hourly or sub-hourly. These devices allow a number of innovative services such as remote metering, dynamic pricing and new ways to perform load forecasting. These technologies help to optimize user electricity consumption through both dynamic price incentives[1] and voluntary participation[2]. The system stress during peak hours will thus be mitigated while users can benefit from their reduced electricity bills or other rewards.

Forecasting of consumption data is crucial to either optimal price strategies or implement demand response (DR). Especially in DR programs, in order to calculate the curtailment effect of price signals or incentives, the electricity consumption without DR signals must be accurately estimated. The difference between the estimated value and the real value during DR events, i.e. peak hours, is then the energy reduction. The focus of this paper is to develop new methods to predict individual consumption.

Several machine learning and data mining approaches have been applied to energy consumption data for prediction purposes. These solutions can be divided into three groups: simple averaging, artificial intelligence(AI) and statistical methods. A literature review is presented in I-B.

In this paper, we investigate autoregression methods for consumption prediction. More specifically, we adopt LASSO to recover the sparsity in historical data. LASSO is a shrinkage and selection method for linear regression[3]. It can recover the sparse solution to coefficient estimation, and in this paper, it can recover the sparsity in autoregressive model parameters. Furthermore, we show that including other users’ data helps with prediction. One intuitive way of including more users into the autoregressive model is by clustering similar users together. However, in this paper we focus on selecting users whose historical data are “predictive” of a specific customer. We thus argue that there exists granger causality between users, which can improve the univariate prediction. Note in this paper, a user or a customer refers to a single household.

The contribution of this paper is two-fold. First, we show that LASSO can recover the recurrent pattern in historical data automatically by shrinking irrelevant coefficients to zero. This greatly improves the interpretation of order selections in autoregressive models. Second, we include one other user according to granger causality tests to improve individual user’s prediction. This aligns with the vector autoregressive (VAR) model; however, in our method the inclusion of the other user is not fixed in advance but rather selected through significance test.

The rest of the paper is organized as follows. Section I-B analyzes related work in short term load forecasting. Section II presents the autoregressive model for time series analysis. Section III introduces the formula for Lasso type estimation. Section IV proceeds granger causality test and introduces the evaluation criteria for prediction methods. Section V details the simulation of several prediction methods and discusses the obtained results. And finally Section VI concludes the paper and draws avenues for future work.

B. Literature review

There exists an extensive literature on short term load forecasting. They can be divided into three groups[4]: simple averaging models, AI models and statistical models. The simplest approach is to employ moving average. Such models make predictions on mean of consumption data from previous similar days[5]. The second category includes models based on AI approaches. This category yields high accuracy at the cost of complexity of the system, which may lead to overfitting[6]. Other drawbacks include difficult parametrization and non-obvious selection of variables, which are difficult to interpret.
In statistical methods, regression models combine several independent features to form a linear function. In [7], the authors build a regression tree model with weather data to predict consumption. Support vector machine is used in [8]. Gaussian process framework for prediction mitigating the uncertainty problem is proposed recently[9]. Besides these methods, times series analysis has also been widely applied to consumption data prediction. An overview can be found in [10].

Another promising statistical method is autoregressive integrated moving average(ARIMA) model[11]. Authors in [12] proposed a vector autoregressive model to include renewables, prices and loads together with sparsity recovery. In [13], the authors applied ARIMA model in short-term load forecasting, and in [14], the authors included electric vehicles into the model. In addition, to extend from linearity to nonlinearity, [15] addresses a mixed model combining ARIMA model to deal with the linear part and neural network with the nonlinear one. In our work, we recover the sparsity for univariate time series and multivariate time series under the framework of autoregressive models. We argue that our methods are simpler and more intuitive that AI methods, thus are more adaptable and interpretable for practical purposes.

II. AUTOREGRESSIVE MODEL

Suppose we regard hourly consumption data as a stochastic time series, the autoregressive time-series model can be expressed as:

\[ y_t = \beta_0 + \sum_{i=1}^{l} \beta_i y_{t-i} + \epsilon_t \] (1)

where \( \{\epsilon_t\} \) is a white noise process, \( \{y_t\} \) is the consumption time series for each individual user, \( \beta_t \) is the coefficient for order (lag order) \( i \) in the autoregressive model, and \( t \) is the time tag in the scale of hours. Note that in this paper we denote a time series data by notation \( \{\bullet_t\} \), where subscript \( t \) refers to the time slots in this time series data. In addition, \( I \) is the number of orders that we want to include in the model. An autoregressive model with maximum order \( I \) is denoted by AR(\( I \)).

In order to estimate the parameters in (1), the usual approach is to apply ordinary least squares (OLS) estimation. We can write \( \{y_{t-1}, y_{t-2}, \ldots, y_{t-I}\}^T \) as a vector denoted by \( X_t \). So the relationship in (1) can be written into canonical form of a linear regression problem:

\[ Y = X\beta + \epsilon \] (2)

where \( Y = [y_t, y_{t+1}, \ldots]^T \), \( X = [1 X_t^T; 1 X_{t+1}^T; \ldots] \), \( \beta = [\beta_0 \beta_1 \ldots]^T \), and \( \epsilon = [\epsilon_t \epsilon_{t+1} \ldots]^T \). The dimension of vectors \( Y, \beta \) and \( \epsilon \) is the length of the vectors, i.e., \( T \), \( I \) and \( T \) respectively. The dimension of a matrix like \( X \) is the number of columns, which implies the number of regressors included in the model.

Under the assumptions for OLS applied in time series data[16], we can apply OLS to the model in (2).

\[ \hat{\beta}_{OLS} = \arg\min_{\beta} \| (Y - X\beta) \|_2^2 \] (3)

where \( \hat{\beta}_{OLS} \) is the estimator for the coefficients in the autoregressive model.

III. SPARSITY RECOVERY IN AUTOREGRESSIVE MODEL

OLS can achieve optimal in-sample performance. Adding more regressors into (2) can always decrease the sum of squared error and better fit the data within the training set. However, when we include too many irrelevant regressors, i.e., when we include too many lag orders in the historical data in (2) , we may be misled by the reduced in-sample bias but ignore the fact of high variance resided in estimators which leads to model overfitting.

Take PG&E dataset as an example. In this particular dataset, hourly consumption data for single households is securely recorded. If we use an AR(3) model for one randomly picked household, it will result in an average in-sample sum of squared error of 1.20, with an average out-of-sample error of 2.08, whereas AR(1) model has an average in-sample sum of squared error of 1.25, together with an average out-of-sample error of 1.91. Thus AR(1) gives better out-of-sample fitting results. If the potential lag orders are up to 10 days, i.e., 240, then an AR(240) model would produce large out-of-sample errors. Thus we need to select the lag orders carefully to avoid model overfitting.

In order to select the most relevant lag orders we propose utilizing the LASSO approach[3]. The LASSO is a shrinkage and selection method for linear regression in (2). It can be expressed as follows:

\[ \hat{\beta}_{LASSO} = \arg\min_{\beta} \frac{1}{2} \| (Y - X\beta) \|_2^2 + \lambda \| \beta \|_1 \] (4)

where \( \lambda \) is a tuning parameter to control the level of sparsity in the solution. For practical purposed, we are using k-fold cross validation to determine the value of \( \lambda \) in our simulations, where \( k \) is either 5 or 10.

In our simulation, LASSO selects both the most recent lag orders and lag orders with intervals of roughly 24 hours, which performs as a combination of simple averaging and AR(\( \cdot \)). As discussed later in Section V, lasso outperforms both simple averaging and AR(1) in terms of relative prediction error, reducing it down by 30%. Furthermore, LASSO also gives a more interpretable results with regards to selected orders. In our simulation, for one electricity user as an example, LASSO selects lag orders as 1, 2, 5, 6, 16, 22, 23, 24, 48, 143, 144, 160, 191, 216, 238, 240. From these orders we can observe a clear behavior pattern of an interval of 24 hours, or a multiple of 24 hours. Unlike simple averaging which fix the lag orders at 24, 48, 72, etc., LASSO will automatically select these orders for each individual based on their respective historical data. In addition, from the result we observe that not every lag order that LASSO picks is a multiple of 24, otherwise we would directly employ simple averaging rather than LASSO.
so LASSO is more adaptive and flexible than simple averaging or AR().

IV. USER PAIRING BY SIGNIFICANCE TEST AND EVALUATION CRITERIA

So far we have considered using historical data of an individual user for its own prediction. One way to leverage the fact that we have many user’s data is to perform vector autoregression(VAR). A VAR model describes the evolution of a set of $k$ variables as a linear function of their historical values. The variables are collected in a $k$ dimensional vector $V_t$ at time $t$. If in our case we want to examine the possible relationship among $k$ users, the regression model in (1) can be extended as:

$$V_t = \mathbf{A}_0 + \sum_{i=1}^{I} \mathbf{A}_i V_{t-i} + \mathbf{\Phi}_i,$$

where $V_t = \{y_{t,1}, y_{t,2}, ..., y_{t,k}\}^T$ is a vector composed of $k$ users’ consumption data at time $t$, $I$ is the number of orders to be considered, $\mathbf{A}_i$’s are the coefficient matrix for orders, and $\mathbf{\Phi}_i$ is the $k$ dimensional white noise.

Again, as stated before, including more users as regressors will reduce the bias but increase the variance for estimators. So the problem here is to select relevant users to be included in (5). Selecting the right user to be included reduces the prediction error compared to AR models. For example, the relative prediction error with the proposed methods will be reduced by around 40% in our simulations if the most relevant user is selected. In this section we use granger causality test to find the best pairing for each user and to construct a two dimensional $V_t$. We also introduce the evaluation criteria to compare different prediction methods.

A. Granger Causality Between Pairs of Users

One way to interpret the pairing between users is to see if the two users are similar enough. Intuitively, this can be accomplished by clustering. However, consider a scenario where two time series have the same exact values for each past time slot and the values at each time slots are i.i.d., then they would be clustered together in almost all methods. Nevertheless, in this case, knowing the history of one time series would not help predict the future values of another time series. On the other hand, if the two time series have similar responses to some common causes but bares certain lag order orders in response, we can use the history of one time series to predict the other. Granger causality defines such spurious causality based on prediction performance.

Granger causality was proposed by Clive Granger, in 1969, to illustrate “causality” between two time series[17]. Now suppose that we have two users’ consumption data, namely $\{y_{t,1}\}$ and $\{y_{t,2}\}$. In the context of Granger causality, the cause is prior to the effect, meaning that if a time series $\{y_{t,1}\}$ is granger causing time series $\{y_{t,2}\}$, then the past observations of $\{y_{t,1}\}$ can help predict current status of $\{y_{t,2}\}$. Obviously granger causality does not imply real causality, in the way that $\{y_{t,1}\}$ and $\{y_{t,2}\}$ can be both generated by some latent factors but $\{y_{t,2}\}$ bares certain lag orders behind $\{y_{t,1}\}$. Still, using historical data of $\{y_{t,1}\}$ helps to predict $\{y_{t,2}\}$.

To test whether $\{y_{t,1}\}$ granger causes $\{y_{t,2}\}$, we consider two autoregressive models to predict $y_{t,2}$:

$$y_{t,2} = \beta_0 + \sum_{i=1}^{I} \beta_{y_{t-i,2}} + \varepsilon_t$$

(6)

$$y_{t,2} = \beta_0 + \sum_{i=1}^{I} \beta_{y_{t-i,1}} + \sum_{j=1}^{J} \gamma_{y_{t-j,1}} + u_t$$

(7)

where $I$ and $J$ are the lags to be included in the model, $\varepsilon_t$ and $u_t$ are the white noises with variance $\sigma^2$.

We say $\{y_{t,1}\}$ granger causes $\{y_{t,2}\}$ if (7) is statistically significantly better than (6). For our application, this definition of causality is justified even if it implies spurious causes in place of real causes. Our focus is on the improvement of prediction accuracy, not on real cause discovery.

Details in testing granger causality can be found in [18]. Here we set $I = J = 1$ to avoid overfitting. We test all possible pairs for each specific user using the $F$ statistics defined in [18] and we reject the null hypothesis if its value is large, i.e., if the associated $p$-value is less than 0.05. We include the most relevant user into the final model by picking up the one user corresponding to the largest $F$ statistics value.

B. Prediction Evaluation Criterion

Once the methods discussed in this paper are proposed, the key issue is whether these methods achieve a better performance. A naive way to evaluate prediction methods is to compare the residual sum of squared error estimated within the testing set. This can be misleading at times because data set in higher scales may have larger residual sum of squared error than data set in smaller scales, whereas methods applied to the former data set can actually have a better performance. One way to solve this problem is to compare the relative error, i.e., the prediction error with respect to the data scale. Here we use the Mean Absolute Percentage Error(MAPE), to capture the relative error:

$$\text{MAPE} = \frac{1}{n} \sum_{t=1}^{n} \left| \frac{y_t - \hat{y}_t}{y_t} \right|$$

(8)

where $y_t$ is the actual value, $\hat{y}_t$ is the predicted value, and $n$ is the number of fitted values.

In case of outliers, we adopt mean curtailing of 0.01 tail and head, or simply use the median to replace the mean value.

V. SIMULATION RESULTS

A. Data Preparation

We use the data from Pacific Gas and Electric Company. It contains anonymized and secure hourly smart meter readings for residential users during a period of one year from 2010 to 2011. Temperature data is retrieved from an online database[19] for the same period. We decompose the consumption data into two parts: 24-hour periodic recurrence and the left-over non-periodic part is perceived as user consumption

978-1-5090-4168-8/16/$31.00 ©2016 IEEE
that reflects behavior. This goal is achieved by averaging for each time unit over all periods in the data[20]. The period is set to be 24 hours. The result of decomposition is shown in Fig.1. The periodic component does not require prediction due to its regularity. The aim is to better predict the left over user behavior, which is the original consumption data minus the periodic recurrence. We also separate the weekday data and weekend data and we focus on weekday data in this paper.

The time series of consumption data after decomposition is stationary. To verify this, we use Augmented Dickey Fuller (ADF) statistics to test the unit root for the autoregressive model. The null hypothesis of ADF is that the data needs to be differenced to make it stationary. It examines this null hypothesis of an ARIMA($k$, 1, 0) process against the stationary ARIMA($k$+1, 0, 0) process, where $k$ is the lag order in the test. Details of ADF can be found in [21]. Besides, in order to validate OLS, some assumptions are made as presented in [18]. We also exclude the temperature effect since it is not significant in our dataset[18].

To perform and evaluate the methods discussed in Section III and in Section IV, we separate the weekday data into training and testing sets. The training set is a sliding window with a length of 50 days, i.e., 1200 hours. The testing set contains one data point which is the next hour following this sliding window.

B. Results for Univariate Time Series Analysis

We compare the simple averaging, univariate autoregression, univariate autoregression with LASSO and multivariate autoregression with granger causality test. The results are shown in Fig.2 and Table I, for 150 users.

As can be seen in Fig.2, LASSO type regression reduces the range of MAPE for these 150 users. The mean is also reduced.

TABLE I: Results for the four methods considered in this paper.

<table>
<thead>
<tr>
<th>Method</th>
<th>mean (MAPE)</th>
<th>AR(1)</th>
<th>LASSO</th>
<th>CauT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Averaging</td>
<td>0.359</td>
<td>0.339</td>
<td>0.225</td>
<td>0.218</td>
</tr>
<tr>
<td>sd (MAPE)</td>
<td>0.237</td>
<td>0.151</td>
<td>0.094</td>
<td>0.090</td>
</tr>
</tbody>
</table>

Particularly, we include as much as 240 lag orders for the LASSO selection, which includes all the historical data for the previous ten days. Taking user No.1 as an example, the LASSO selects 16 non zero lag orders, according to 10-fold cross validation with a sequence of decreasing {$\lambda_k$}. The lag orders that the LASSO picks are 1, 2, 5, 6, 16, 22, 23, 24, 48, 143, 144, 160, 191, 216, 238, 240. This pattern reflects that LASSO not only selects the most recent lag orders (which is similar to AR(1)), but the lag orders roughly at interval of one day, i.e., 24 hours as well (which is similar to simple averaging). The coefficients for the lag orders also have different scales. In the example of user No.1, the most recent lag orders given the largest coefficient ($\beta_1 = 0.259$). The second largest coefficient is give to lag order 24 ($\beta_{24} = 0.187$). The rest of the coefficients are scaled between around 0.01 to 0.06.

Above all, multivariate autoregression with granger causality test achieves the best performance among the four methods. It can be seen that including a predictive user can perform a good estimation as well as including most important lag orders in its own historical data. A comparison of the four methods is summarized in Table I, where CauT stands for test based on granger causality.

C. Aggregation Over Users

In the above analysis we consider each individual’s consumption data as a stochastic time series data. Alternatively, we can aggregate multiple users together and treat the aggregated data as a time series data. The concept of aggregation of forecasting and baselines was explored in [9] and [22]. Intuitively, the more users we aggregate, the less the noise we will get. This is because the noise term is assumed to be zero mean and finite variance. With aggregation, the fluctuation of the individual-level noise will gradually cancel out with each other. Thus, aggregating users will reduce the overall error and will let to better prediction, but at the cost of sacrificing individual consumption information. Fig.3 shows user aggregation tradeoff.
500 users, we can at most generate 3 aggregated consumption patterns. The result is not robust due to the limited size of the dataset. Based on the dataset, for aggregation level at 500 users, we can at most generate 3 aggregated consumption data from our data set through randomized grouping. Thus pairing for these consumption data is not as evident as in the individual consumption data analysis.

VI. CONCLUSION AND FUTURE WORK

In this paper, we pose and analyze consumption prediction methods based on autoregressive models. For individual users, we treat their consumption data as univariate time series and adopt LASSO to recover the sparsity in lag orders. The LASSO method selects the most recent lag orders and important lag orders with multiples of 24 hours, which reveals the consumption pattern. Furthermore, we extend the idea from individual users to possible relations between users. We test granger causality between pairs of users, and treat the two-user autoregression problem as a vector autoregressive model with lag order one. The test of granger causality can be seen as a possible improvement to AR(1). The simulation results show that LASSO selection outperforms both simple averaging method and autoregressive model with lag order one for individual consumption prediction. In addition, adding the lag order one historical data from another relevant user will enhance the prediction accuracy, just as adding most important lag orders from its own historical data. These prediction methods can help compare the user behavior with and without demand response incentives. The proposed methods can also be easily applied towards mid-term or long-term forecasting when there are some recurrent behavioral patterns resided in the data. In the future, we will use the results in this paper to investigate the demand response effect for users.

REFERENCES