UNIVERSITY OF CALIFORNIA SAN DIEGO

Short-Term Electric Load Prediction Using Multiple Linear Regression Method

A thesis submitted in partial satisfaction of the
requirements for the degree
Master of Science

in

Electrical Engineering (with a specialization in Communication Theory & Systems)

by

Jun Tae Kim

Committee in charge:

Professor Ramesh R. Rao, Chair
Professor William S. Hodgkiss
Professor Ken Kreutz-Delgado

2018

The thesis of Jun Tae Kim is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

____

____

____

Chair

University of California San Diego

2018

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

VITA

2003	B.S. in Engineering, Yonsei University, South Korea
2003 - 2015	Employee, Korea Electric Power Corporation, South Korea
2015 -	Deputy General Manager, Korea Electric Power Corporation, South Korea
2018	M.S. in Electrical Engineering, University of California San Diego, USA

ABSTRACT OF THE THESIS

Short-Term Electric Load Prediction Using Multiple Linear Regression Method

by

Jun Tae Kim

Master of Science in Electrical Engineering (with a specialization in Communication Theory & Systems)

University of California San Diego, 2018

Professor Ramesh R. Rao, Chair

This paper provides new techniques to predict electric loads using the Multiple Linear Regression (MLR) model, which adopts a statistical approach that assumes the past load and weather data have information for forecasting the target load. Since the conventional general MLR prediction performance can be degraded by seasonal effects, we propose new MLR techniques to improve the prediction performance. We have found the performance of the proposed MLR can be further improved by solving the weighted least squares problem and clustering the training set. Additionally, we compare the prediction performances of these techniques to determine the best one. Our argument will be demonstrated by two case studies with real electric and weather data.

Chapter 1

Introduction

Electric load forecasting is most essential for efficient power grid operation and management. Electrical energy as consumer goods requires maximal consumption due to its difficulty in storage. Therefore, electric load forecasting is key to reducing unnecessary power generation, which prevents wasting resources. Furthermore, it can be used for Transmission and Distribution (T&D) planning, Demand Side Management (DSM), and intelligent trade in the competitive energy market. Load forecasting can be classified into three types according to its evaluation time range and forecasting purpose.

- Long-term Load Forecasting (LTLF) – Forecasting the peak electric demand of multiple years, which is essential for power capacity planning [MBB05].
- Mid-term Load Forecasting (MTLF) – Forecasting the hourly load of several weeks up to one year, which is needed for power system maintenance scheduling [EMD06].
- Short-term Load Forecasting (STLF) - Forecasting the hourly load up to 24 hours, which is implemented for power system operation and DSM.

STLF allows operators to make critical decisions for controlling power systems, maintaining grid stability, and reducing excessive power generation or waste of resources. Moghram

[MR89] introduced some classical and typical methods of STLF, which consist of the stochastic time series method, state space method, knowledge-based approach, and multiple linear regression (MLR) method.

The electric load is formulated as a function of the past observed load value in the listed methods above except for the knowledge-based approach. The examples of stochastic time series method are autoregressive models, moving-average models, and autoregressive moving-average models. The state space method employs the Kalman filter. Since all of them use predicted load values as input to estimate their future seqential load values for multi-step forecasting, the estimation error tends to propagate and become more serious as the the prediction steps increase. The knowledge-based approach implements various methods and is developed with the expert system and fuzzy theory. After Czernichow [CPIC96] utilized artificial neural networks (ANN) for load forecasting in his paper, many researchers have paid attention to this field. In recent years, ANN has become one of the major techniques for load forecasting and has been applied to various industrial fields of energy utility companies [KARM98]. However, this method requires large amounts of data for learning and has overfitting problems caused by an ambiguous criterion for implementing and stopping iterative functions [HPS01]. The MLR method is one of the most popular statistical methods of load forecasting, on which many studies are still being conducted. It is easy to set up a mathematical model and interpret the behavoral characteristics of the estimated load.

In this paper, we introduce a new MLR method based on the former papers utilizing MLR and formulate the load with past load and weather information. Unlike the conventional general MLR which selects the observed load data based on a specific time and learns the patterns, this new approach selects the observed load data based on a specific reference load. We try to address some problems in order to apply this new proposed MLR (PMLR) and compare prediction performances in exploration of how much we can improve them with new techniques. Some case studies are conducted to verify our argument with real electric and weather data.

The rest of this paper is organized as follows: Chapter 2 outlines the theoretical backgrounds and Chapter 3 illustrates the detailed methodology to implement MLR predictions with proposals of application methods to improve prediction performance. The results of the case studies with real electric load data are presented and analyzed in Chapter 4. Chapter 5 concludes this paper.

Chapter 2

Background

2.1 Multiple Linear Regression

In the MLR method, the load is described as a linear combination of explanatory variables that influence the load such that

$$y = c_0 x_0 + c_1 x_1 + \ \cdots \ + c_n x_n + e = \vec{c} \cdot \vec{x} + e, \tag{2.1}$$

where y is the electrical load, c_j is the explanatory variable, x_j is the unknown regression coefficient for $j \in [0, n]$, and e is the modeling error. The row vector $\vec{c} = [c_0, \ c_1, \ \cdots, \ c_n]$ contains dependent variables which are affecting on the load and the column vector $\vec{x} = [x_0, \ x_1, \ \cdots, \ x_n]^t$ consists of unknown coefficients where a^t refers to a transposition of a. If we choose some observed data from a training set and regard them as being formulated by Eq. (2.1), we can set up an MLR model as an overdetermined system. Let y_i denote the i-th observed load and and c_{ij} denote the j-th component of the ith observed condition vector $\vec{c}_i$ for m observed electric load data. The i-th estimated load for $i \in [1,\ m]$ is given by

$$y_i = \sum_{j=0}^{n} c_{ij} x_j \ \text{ for } \ i \in [1,\ m]. \tag{2.2}$$

Eq. (2.2) has m equations of y_i and $\vec{c}_i$ with $n+1$ unknown coefficients $x_0,\ \cdots,\ x_n$ with $m > n+1$ and can be written in the vector and matrix forms as follows.

$$\vec{y} = C \cdot \vec{x}, \tag{2.3}$$

where

$$\vec{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}, \ C = \begin{bmatrix} c_{10} & c_{11} & \cdots & c_{1n} \\ c_{20} & c_{21} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m0} & c_{m1} & \cdots & c_{mn} \end{bmatrix}$$

This equation usually has no solution with $m > n+1$. Instead of finding a unique solution, we need to find the coefficient $\vec{x}$ that best minimizes the squares of error between the estimated and actual loads to fit Eq. (2.3). This method is well known as the Ordinary Least Squares (OLS) problem, which has a unique solution as follows, given that the $n+1$ columns of matrix C are linearly independent.

$$\begin{aligned} \vec{x}^* &= \arg\min_x \|\vec{y} - C \cdot \vec{x}\|^2 \\ &= (C^t \cdot C)^{-1} \cdot C^t \cdot \vec{y} \end{aligned} \tag{2.4}$$

With this optimized $\vec{x}^*$ in Eq. (2.4) and condition vector $\vec{c}$, we can predict the next state load $\hat{y}$ as follows.

$$\hat{y} = \vec{c} \cdot \vec{x}^* \tag{2.5}$$

This estimated load value will be compared to that of the actual load in order to measure the prediction performance later.

2.2 Weighted Least Squares

When we solve an ordinary least squares (OLS) problem to optimize the coefficient vector, we assume that all of the observed load data have the same error variance. But usually, the observation error variance might be different on each observation. We may have a small error variance on a certain observation, which means that the observation might be precise, thus making the data highly reliable. On the other hand, we may have large error variances on the other observation, which indicate that the observation might be inaccurate, thus making the data less believable. If we know the weights for all m observed data, the squared error in Eq. (2.4) should be modified as follows considering these weights. Let $S(x)$ denote the weighted squared error function, w_i denote the weight of i-th observation, and W denote the diagonal matrix of $diag(w_i)$, where $i = 1,\ 2,\ \cdots,\ m$. Then $S(x)$ is given by

$$S(x) = \sum_i w_i \cdot \left\| y_i - \sum_j c_{ij} x_j \right\|^2 = \left\| W^{1/2}(\vec{y} - C \cdot \vec{x}) \right\|^2. \tag{2.6}$$

Eq. (2.6) becomes a weighted least squares (WLS) problem and the optimized $\vec{x}^*$ minimizing the weighted squared error function $S(x)$ can be calculated as

$$\vec{x}^* = \arg\min_{\vec{x}} S(x) = (C^t \cdot W \cdot C)^{-1} C^t \cdot W \cdot \vec{y}. \tag{2.7}$$

This WLS problem can only be applied in rare cases where we know how to establish the weight for each observation. However in most cases, we do not know the structure of the weight matrix W. In the next chapter, we will introduce several methods for establishing the weight matrix W, which may improve prediction performance.

2.3 The *k*-means Clustering Algorithm

Another way to improve prediction performance is to divide the training set into several clusters to reduce the variance of the dataset. If we predict the load during the summer by only using the summer training set, the prediction performance may increase, while also reducing the amount of computation. To do this, we need to divide the training set by some criteria, which we will determine by k-means clustering. This algorithm aims to group n observed data into k clusters in which each variance of distances from the center of clusters is minimized. Given a set of n data ($\vec{x}_1, \vec{x}_2, ..., \vec{x}_n$), which are d-dimensional vectors, the k-means algorithm computes n data objects into k sets $S = \{S_1,\ S_2,\ \cdots,\ S_k\}$ by maximizing the cohesion between the objects in each set. In other words, when $\vec{\mu}_i$ is the center point of the set S_i, the goal of this algorithm is to find the set S that minimizes the sum of squares of the distances between the center points of each set and the objects in the set. The equation is expressed as follows.

$$\arg\min_{S} \sum_{i=1}^{k} \sum_{\vec{x} \in S_i} ||\vec{x} - \vec{\mu}_i||^2 \tag{2.8}$$

The algorithm sets the initial $\vec{\mu}_i$ and repeats the next two steps.

- Assignment step: Assign each observation to the cluster whose mean $\vec{\mu}_i$ is nearest to the observation such that

$$S_i{}^{(t)} = \left\{ \vec{x}_p : \left\| \vec{x}_p - \vec{\mu}_i{}^{(t)} \right\|^2 \le \left\| \vec{x}_p - \vec{\mu}_j{}^{(t)} \right\|^2 \ \forall j,\ 1 \le j \le k \right\},$$

- Update step: Update the new $\vec{\mu}_i$ to be the centroids of the observations in the new clusters such that

$$\vec{\mu}_i^{(t+1)} = \frac{1}{\left|S_i{}^{(t)}\right|} \sum_{\vec{x}_j \in S_i{}^{(t)}} \vec{x}_j,$$

  where $|A|$ denotes the number of elements in set A.

The algorithm stops iterating when the assigned cluster no longer changes. The k-means algorithm has limitations such as the initial cluster setting problem and no guarantee of a global minimum. Some techniques are developed to overcome these limitations, but we will not discuss them in this paper. We will try to find the optimal k with some experiments using the `kmeans` function provided by MATLAB.

Chapter 3

Methodology

3.1 Overall Description

The condition vector $\vec{c}$ has been defined in various ways in the previous extant literatures using MLR. One of these examples is the use of weather information. Some researchers defined the condition vector $\vec{c}$ using various weather information such as temperature of the previous hour, average temperature of the previous 24 hours, temperature difference of a certain time interval, square of this difference, humidity, wind speed and so on. Rahman [RH93] defined some conditional variables for each season considering the seasonal effect such as using only temperature variables in the summer and using temperature and chill factors in the winter. And the load was modeled by dividing a day into six sections and considering the hourly effect of the day. On the other hand, some time series models such as autoregressive, autoregressive moving average, and the Kalman filter employ past load data to set up conditional variables. To establish the condition vector $\vec{c}$ is the most important step in forecasting the load, since it has a significant impact on the forecasting performance depending on which elements are inserted or excluded. Generally speaking, the past load and temperature values are the greatest influential factors in load forecasting.

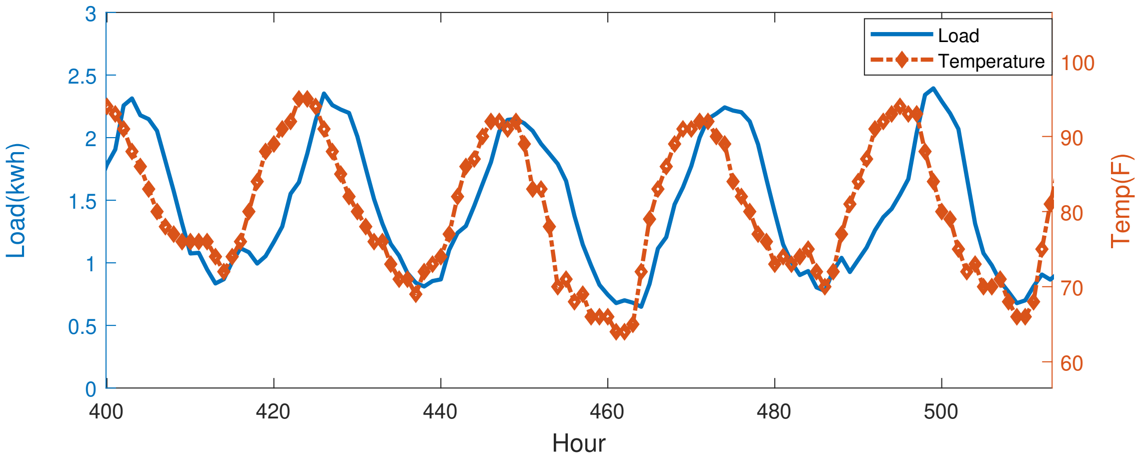


Figure 3.1: Average residential load (Blue) and wet bulb temperature (Red) on September 2015 in Austin, Texas, USA.

Fig. 3.1 shows the average residential load and temperature during September in the Austin, Texas, USA area. The load and temperature have a periodicity on a daily basis and they are highly correlated with each other. It is apparent that the present load is related to the previous one, such as an hour ago or 24 hours ago. The present load also depends on the transition amount of the recent load and temperature. Applying this fact to Eq. (2.1), the load can be formulated with an unknown coefficient vector $\vec{x}$ and the condition vector $\vec{c}$, which contains the past load and temperature. Let $y(kT)$ denote the electric load at time kT with the observation time interval T, c_j denoting the j-th component of condition vector $\vec{c}_k$, which consists of previous loads dependent on $f_1 \sim f_n$ and temperature dependent on $g_1 \sim g_m$, and x_j denote the j-th component of an unknown coefficient vector $\vec{x}_k$. Then, $y(kT)$ is given by

$$\begin{aligned} y(kT) &= x_0 + c_1x_1 + c_2x_2 + \cdots + c_lx_l + e \\ &= x_0 + f_1x_1 + \cdots + f_nx_n + g_1x_{n+1} + \cdots + g_mx_{n+m} + e \\ &= [1,\ f_1,\ \cdots, f_n,\ g_1,\ \cdots,\ g_m] \cdot [x_0,\ x_1,\ \cdots,\ x_{n+m}]^t + e = \vec{c}_k \cdot \vec{x}_k + e. \end{aligned} \tag{3.1}$$

The f is a load-related function while g is a temperature-related function, both determined by how the MLR method is implemented. Since the components number of $\vec{c}_k$ affects the prediction performance, $\vec{c}_k$ is equally established including one constant component, three past load dependent components, and three temperature dependent components for different MLR methods. This paper focuses more on the observation method rather than how to design the condition vector $\vec{c}_k$ of each MLR. The components of the condition vector can be defined by the observation and regression method differently to forecast successive 24-hour load points. We will choose the observed data with different methods as follows depending on each MLR prediction for a successive 24-hour prediction.

- General MLR (GMLR) – We observe some past loads at the same time of the target time and collect them to make the observed data vector and condition matrix.
- Proposed MLR (PMLR) – We set up a reference load prior to the target time and collect past loads which are close to the reference load to make the observed data vector and condition matrix.

We set up the condition vector in the GMLR predicion with the loads and temperatures of a specific past time, which is proposed by Hamadi [HHS04] in his paper. On the other hand, we establish the condition vector differently in the PMLR prediction with a specific reference load and change the amount of past loads and temperatures. With the aforementioned observed vector and condition matrix, we can calculate the optimized coefficient vector and predict the target load with the condition vector as shown Eq. (2.4) and Eq. (2.5).

3.2 General Multiple Linear Regression

In the GMLR, we establish the condition vector $\vec{c}_k$ based on Eq. (3.1) such that

$$\begin{aligned}\vec{c}_k = [1,\ & y((k-1)T),\ y((k-24)T),\ y((k-25)T), \\ & h((k-1)T),\ h((k-24)T),\ h((k-25)T)], \end{aligned} \tag{3.2}$$

where $y((k-k_i)T)$ denotes the past load and $h((k-k_i)T)$ denotes the past temperature. If we compare Eq. (3.2) with Eq. (3.1), the past three loads at a specific time are the load-related function f and the past three temperatures at a certain time are the temperature-related function g. Typically, the correlation with the target load $y(kT)$ is the most important factor for selecting the components of the condition vector $\vec{c}_k$, but additional factors should be considered for a continuous 24-hour prediction. Since the 1-hour-ago load $y((k-1)T)$ has the strongest correlation with the current load $y(kT)$ and so is therefore the most powerful explanatory variable, it is naturally one of the components of $\vec{c}_k$. Likewise, the 2-hours-ago load $y((k-2)T)$ and the 3-hours-ago load $y((k-3)T)$ also have strong correlations with $y(kT)$ and should be included in $\vec{c}_k$. However, in a continuous 24-hour prediction, they can matter. If all of $y((k-1)T)$, $y((k-2)T)$, and $y((k-3)T)$ are used in $\vec{c}_k$, the previously predicted values must be iteratively applied to set up the $\vec{c}_k$ for the next prediction. In this case, since the prediction error may become too large, loads over 24 hours such as $y((k-24)T)$ and $y((k-25)T)$ are used with $y((k-1)T)$ instead of $y((k-2)T)$ and $y((k-3)T)$. The temperature h in the condition vector $\vec{c}_k$ can be implemented assuming that the temperature data of the target day is given by the weather forecast.

In addition, we need to decide how we choose an observed data set in order to implement the prediction based on Eq. (2.3). In the GMLR, we observe the past loads at the same time of each day and collect them to establish an observed load vector $\vec{y}$ and a condition matrix C. Let $y(kT)$ denote a target load which we want to predict, $\vec{c}_k$ denote the condition vector at time kT, and $\vec{x}_k$ denote the unknown coefficient vector which is dependent on time kT for the target load

$y(kT)$. Assuming that we observe the loads of the past m days, then the observed load vector $\vec{y}_k$ and the condition matrix C_k can be found as follows.

$$\vec{y}_k = \begin{bmatrix} y((k-24)T) \\ y((k-48)T) \\ \vdots \\ y((k-24m)T) \end{bmatrix}, C_k = \begin{bmatrix} \vec{c}_{k-24} \\ \vec{c}_{k-48} \\ \vdots \\ \vec{c}_{k-24m} \end{bmatrix} \tag{3.3}$$

With this $\vec{y}_k$ and C_k, we can calculate the optimized $\vec{x}_k^*$ by solving the least square problem in Eq. (2.4) such that

$$\vec{x}_k^* = (C_k^t \cdot C_k)^{-1} C_k^t \cdot \vec{y}_k. \tag{3.4}$$

We can also predict the successive 24 loads as follows.

$$\hat{y}(kT) = \vec{c}_k \cdot \vec{x}_k^* \quad where\, k = 0,\ 1,\ \cdots,\ 23. \tag{3.5}$$

The $\hat{y}(0)$ for $k = 0$ can be estimated directly from $\vec{c}_0$ and $\vec{x}_0^*$ based on Eq. (3.5), but we need the predicted value $\hat{y}(0)$ as an input for setting up $\vec{c}_1$, which we need for the next prediction. For a successive 24-hour prediction, we need to use the previous estimated load as an input to establish $\vec{c}_k$ for the $\hat{y}(kT)$ for $k = 1, ..., 23$. Fig. 3.2 shows a brief overview of the iterative GMLR procedure for successive 24-hour prediction described so far.

3.3 Proposed Multiple Linear Regression

In the GMLR, we collect the observed data on a time basis and predict the target load. Since this GMLR method depends on the observation period, the error tends to increase when a specific variable suddenly changes during the observation period. To reduce this time effect, we

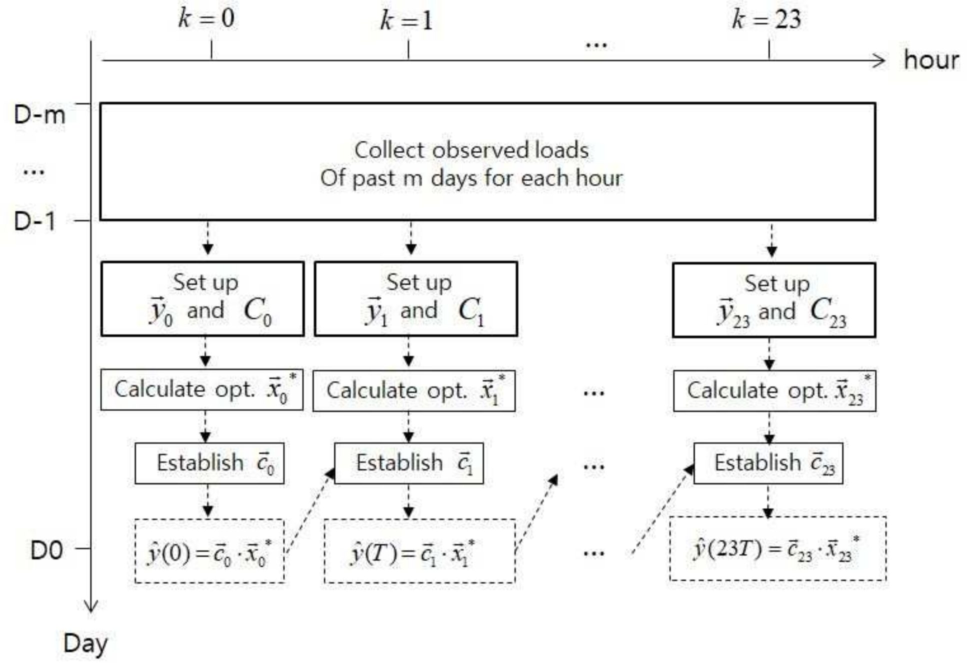

$k=0$
$k=1$
...
$k=23$
hour
D-m
...
D-1
D0
Day
Collect observed loads
Of past m days for each hour
Set up $\vec{y}_0$ and C_0
Set up $\vec{y}_1$ and C_1
Set up $\vec{y}_{23}$ and C_{23}
Calculate opt. $\vec{x}_0^*$
Calculate opt. $\vec{x}_1^*$
...
Calculate opt. $\vec{x}_{23}^*$
Establish $\vec{c}_0$
Establish $\vec{c}_1$
...
Establish $\vec{c}_{23}$
$\hat{y}(0)=\vec{c}_0\cdot\vec{x}_0^*$
$\hat{y}(T)=\vec{c}_1\cdot\vec{x}_1^*$
...
$\hat{y}(23T)=\vec{c}_{23}\cdot\vec{x}_{23}^*$

Figure 3.2: Brief overview of a 24-hour prediction using the interative GMLR method.

propose a new method of MLR (PMLR), which chooses the observed data on a past specific load basis, not time basis. In this PMLR method, we need to establish another condition vector $\vec{c}_k$, which is different from the one used in GMLR in Eq. (3.2) for a successive 24-hour prediction. If the condition vector $\vec{c}_k$ contains a load value of less than 24 hours, it is necessary to use the previous estimation result iteratively for the next prediction, which is similar to GMLR. This iterative method in PMLR leads to an unexpected large error that is much more serious than in GMLR. To avoid this large error, we set up the 24-hours-ago load as a reference load. Then, we can predict the target load $y(kT)$ directly without an iterative method. In this method, the condition vector $\vec{c}_k$ can be proposed as follows.

$$\begin{aligned}\vec{c}_k = [\,&1, y((k-25)T), y((k-25)T)-y((k-26)T), y((k-25)T)-y((k-49)T),\\ &h((k-1)T)-h((k-2)T), h((k-2)T)-h((k-3)T), h((k-1)T)-h((k-25)T)\,] \quad (3.6)\end{aligned}$$

In Eq. (3.6), we don't need to apply the previous estimated load iteratively for next state prediction when we predict successive 24-hour-loads, since $\vec{c}_k$ contains no load from less than 24 hours ago. It is assumed that the past load at a specific time $y((k-25)T)$, the load fluctuation Δy, and the temperature change Δh affect the estimated load $y(kT)$. The load fluctuation Δy and the temperature change Δh are taken into account due to the effects of hourly or daily changes.

In the PMLR, we gather information from the previous load by collecting the observed data, which is same or close to the reference load, and infer the target load from the information of the chosen observed data. In other words, the PMLR is different from the GMLR in the way we choose the observed data that is needed to set up $\vec{y}$ and C. Let $y\left((k-24)T\right)$ denote the 24-hours-ago load as a reference load (previous state) for a target load $y(kT)$ (next state), and $l((k_i-24)T)$ denote i-th chosen past load, which is satisfying $l((k_i-24)T) = y((k-24)T) = \alpha_k$. Assuming the reference $y\left((k-24)T\right)$ is α_k, we can collect the points of $l\left((k_i-24)T\right) = \alpha_k$ and estimate $y(kT)$ using the correlation of $l\left((k_i-24)T\right)$ and $l(k_iT)$. Fig. 3.3 illustrates how we can choose such points from the correlation graph of previous and next state loads in the training set when α_k is set to 0.4. The transition from $l\left((k_i-24)T\right)$ to $l(k_iT)$ makes it possible to infer the $y(kT)$ from $y\left((k-24)T\right)$. More specifically, assuming m points that satisfy $l\left((k_i-24)T\right) = \alpha_k$, the chosen observed data set Y can be defined as follows.

$$Y = \{y_i = l(k_iT) |\, l((k_i-24)T) = y((k-24)T) = \alpha_k,\ for\ i \in [1,\ m]\}$$

And the observed data vector $\vec{y}_k$ and the condition matrix C_k at time kT will be set to as follows.

$$\vec{y}_k = \begin{bmatrix} y_1 = l(k_1T) \\ y_2 = l(k_2T) \\ \vdots \\ y_m = l(k_mT) \end{bmatrix},\ C_k = \begin{bmatrix} \vec{c}_1 : \ Past\ condition\ vector\ of\ y_1 \\ \vec{c}_2 : \ Past\ condition\ vector\ of\ y_2 \\ \vdots \\ \vec{c}_m : \ Past\ condition\ vector\ of\ y_m \end{bmatrix} \tag{3.7}$$

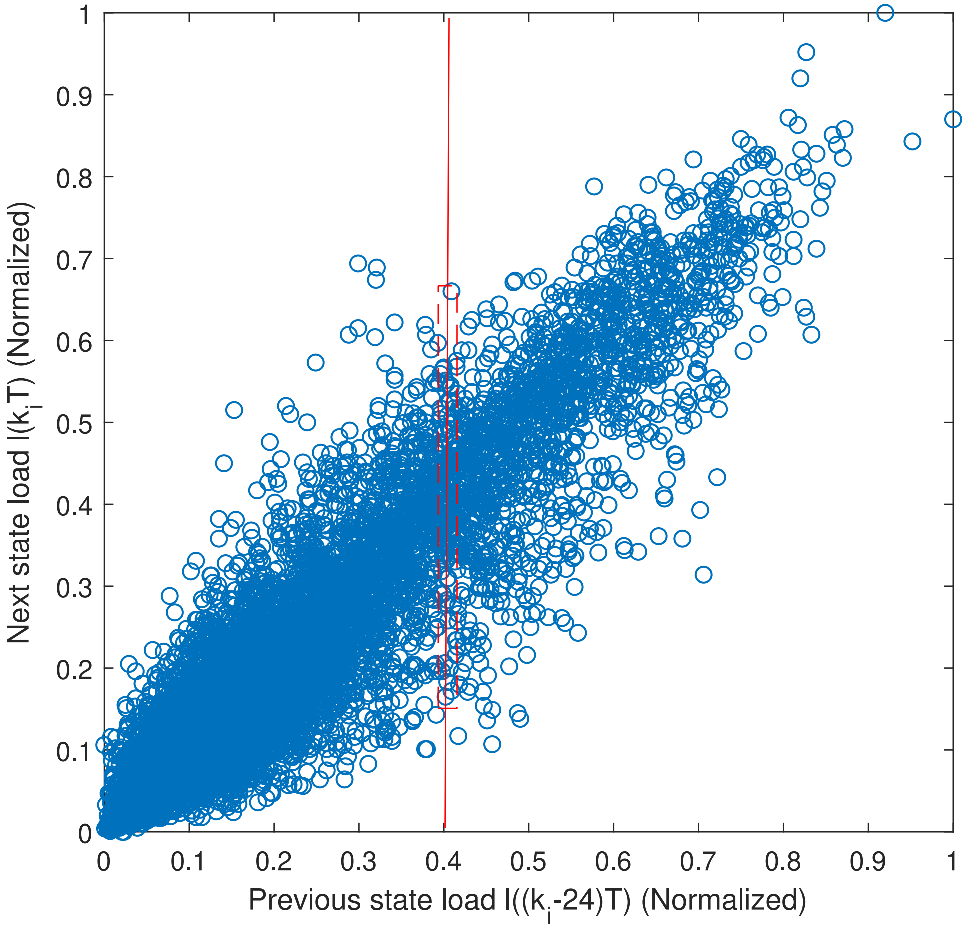


Figure 3.3: Correlation graph between previous state and next state loads in the training set to determine the observed loads satisfying $l((k_i-24)T) = y((k-24)T) = \alpha_k$ when α_k is set to 0.4.

If we find enough m points for the vector $\vec{y}_k$ and matrix C_k such that m is bigger than the unknowns from the correlation graph in Fig. 3.3, the optimized $\vec{x}_k^*$ can be easily achieved as follows by solving Eq. (2.4).

$$\vec{x}_k^* = (C_k^t \cdot C_k)^{-1} C_k^t \cdot \vec{y}_k. \tag{3.8}$$

Eq. (3.8) is same as Eq. (3.4) in this format, but the contents are different from each other since $\vec{y}_k$ and C_k in each equation are chosen with different methods. We can also predict incoming successive 24 loads with the previous 24 loads as follows.

$$\hat{y}(kT) = \vec{c}_k \cdot \vec{x}_k^* \quad where\ k = 0,\ 1,\ \cdots,\ 23. \tag{3.9}$$

Fig. 3.4 shows the successive 24-hour prediction using the PMLR with Eq. (3.9). While we should use the previous predicted value recursively as an input for the next prediction in the GMLR, we can predict the successive 24 loads without iteration in this PMLR.

3.3.1 Approximately Adaptive Searching

Basically, we assume that the number of rows of matrix C is bigger than the dimension of the unknown vector $\vec{x}$ when computing the optimized $\vec{x}^*$ in Eq. (2.4). Let d denote the dimension of the vector $\vec{x}$ and m denote the number of rows of C. This can be solved only if the number of equations, m, is greater than the number of unknowns d. If m is smaller than d, we cannot compute the inverse matrix of $C^t \cdot C$ since $C^t \cdot C$ is a singular. The first problem of the proposed method occurs when the elements of observed data set Y is small. In case that the reference load is at a low density position, the number of equations can be smaller than the dimension of the unknown coefficient vector. Fig. 3.5 shows this case. When α_k is 0.45 for the reference load $y((k-24)T)$, there are 4 points of $l((ki-24)T) = 0.45$. In this case, the least squares equation

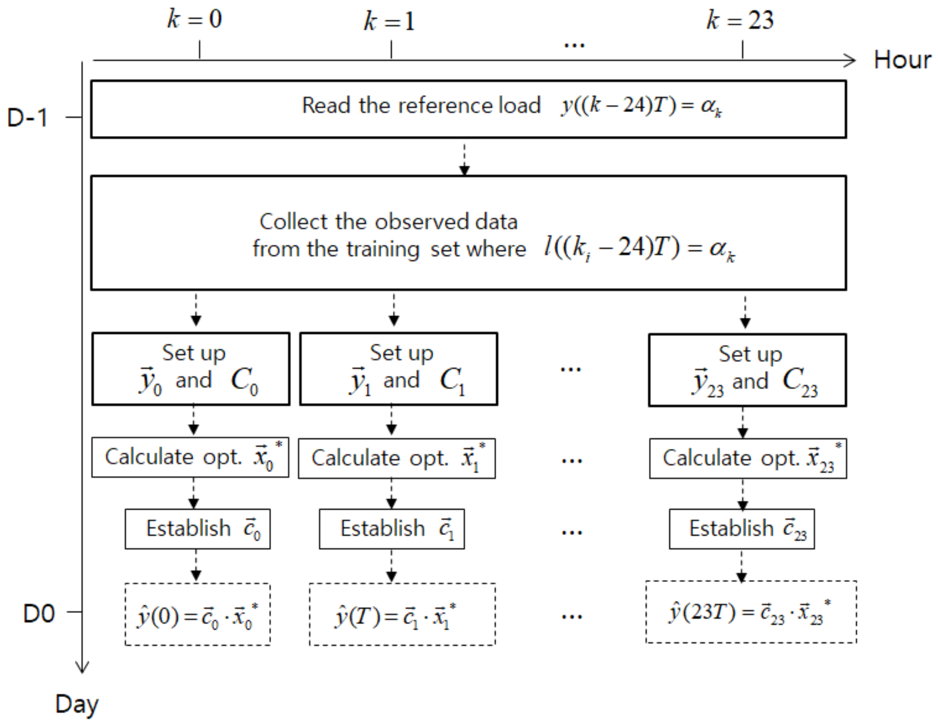

$k=0$
$k=1$
…
$k=23$
Hour
D-1
Read the reference load $y((k-24)T)=\alpha_k$
Collect the observed data
from the training set where $l((k_i-24)T)=\alpha_k$
Set up
$\vec{y}_0$ and C_0
Set up
$\vec{y}_1$ and C_1
…
Set up
$\vec{y}_{23}$ and C_{23}
Calculate opt. $\vec{x}_0^*$
Calculate opt. $\vec{x}_1^*$
…
Calculate opt. $\vec{x}_{23}^*$
Establish $\vec{c}_0$
Establish $\vec{c}_1$
…
Establish $\vec{c}_{23}$
D0
$\hat{y}(0)=\vec{c}_0\cdot\vec{x}_0^*$
$\hat{y}(T)=\vec{c}_1\cdot\vec{x}_1^*$
…
$\hat{y}(23T)=\vec{c}_{23}\cdot\vec{x}_{23}^*$
Day

Figure 3.4: Brief overview of a 24-hour prediction using the PMLR

$\vec{y}_k = C_k \cdot \vec{x}_k$ will be established as follows.

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} c_{10} & c_{11} & \cdots & c_{16} \\ c_{20} & c_{21} & \cdots & c_{26} \\ c_{30} & c_{31} & \cdots & c_{36} \\ c_{40} & c_{41} & \cdots & c_{46} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_6 \end{bmatrix} \tag{3.10}$$

Since the dimension of the unknown vector $\vec{x}_k$ is 7 as in Eq. (3.10), m should be larger than 7 in order to calculate $\vec{x}_k^*$ in Eq. (3.8). However, since m is smaller than d ($4 < 7$), the unknown vector $\vec{x}_k^*$ has infinite solutions, which is useless for our prediction.

To solve this problem, we introduce the approximately adaptive searching method to obtain more points by accepting similar points instead of only using the same exact points. Since the proposed method tries to learn past experiences based on the information specialized by

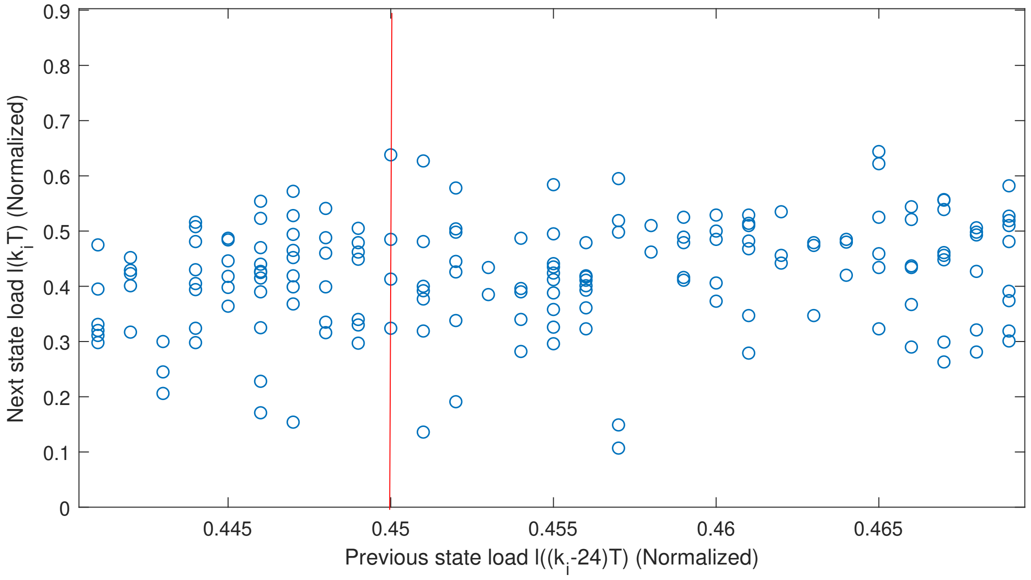


Figure 3.5: Example of an insufficient number of chosen points when α_k is set to 0.45.

the reference, it is also meaningful to employ approximations around the reference point. If we take those approximations, we can increase m more. Fig. 3.6 shows the approximately adaptive searching method for a small m near $l((k_i-24)T) = \alpha_k$. If we adopt the similar points of $\alpha_k - \varepsilon_k \leq l((k_i-24)T) \leq \alpha_k - \varepsilon_k$, where ε_k is an adaptively increasing searching interval, we can obtain many more training points compared to the 4 points in Fig. 3.5. In Fig. 3.6, we accept the 18 points of $0.449 \leq l((k_i-24)T) \leq 0.451$ as the chosen observed points for the reference $\alpha_k = 0.45$. Then, Eq. (3.10) can be modified as follows.

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{18} \end{bmatrix} = \begin{bmatrix} c_{10} & c_{11} & \cdots & c_{16} \\ c_{20} & c_{21} & \cdots & c_{26} \\ \vdots & \vdots & & \vdots \\ c_{180} & c_{181} & \cdots & c_{186} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_6 \end{bmatrix} \tag{3.11}$$

Since the number of equations is bigger than the unknowns ($18 > 7$) in Eq. (3.11), we can find the optimized $\vec{x}_k^*$ in Eq. (3.8). When we have insufficient observed points satisfying

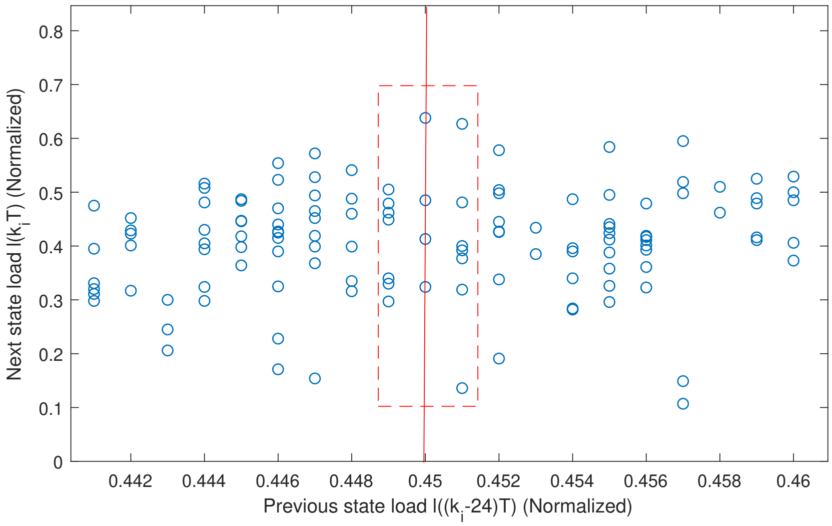


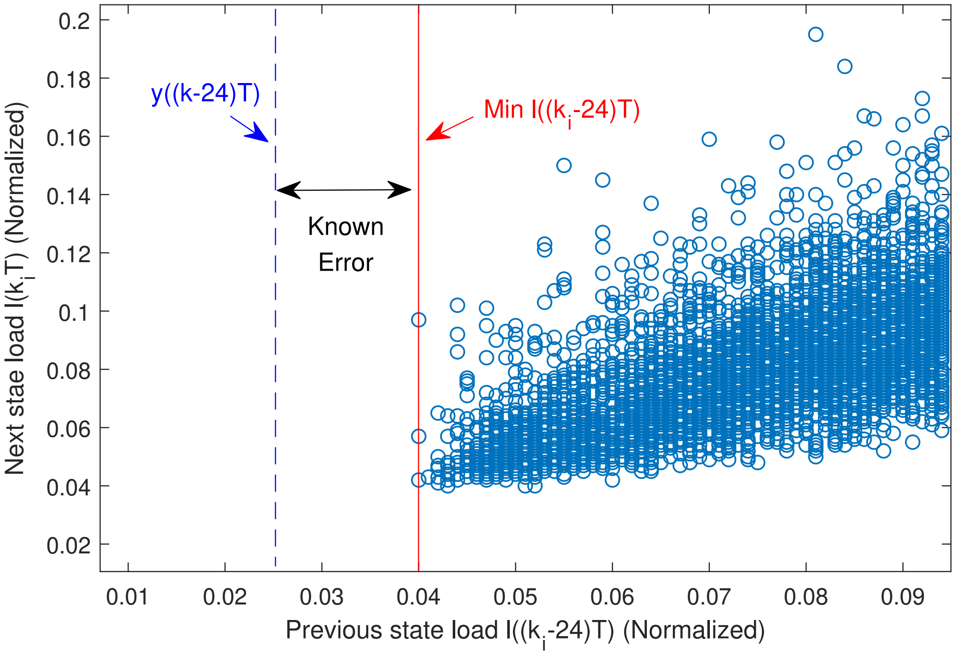
Figure 3.6: Illustration of the approximately adaptive searching technique when α_k is 0.45.

$l((k_i - 24)T) = \alpha_k$, we can obtain more training points, which are enough to calculate $\vec{x}_k^*$ by adaptively increasing the searching interval.

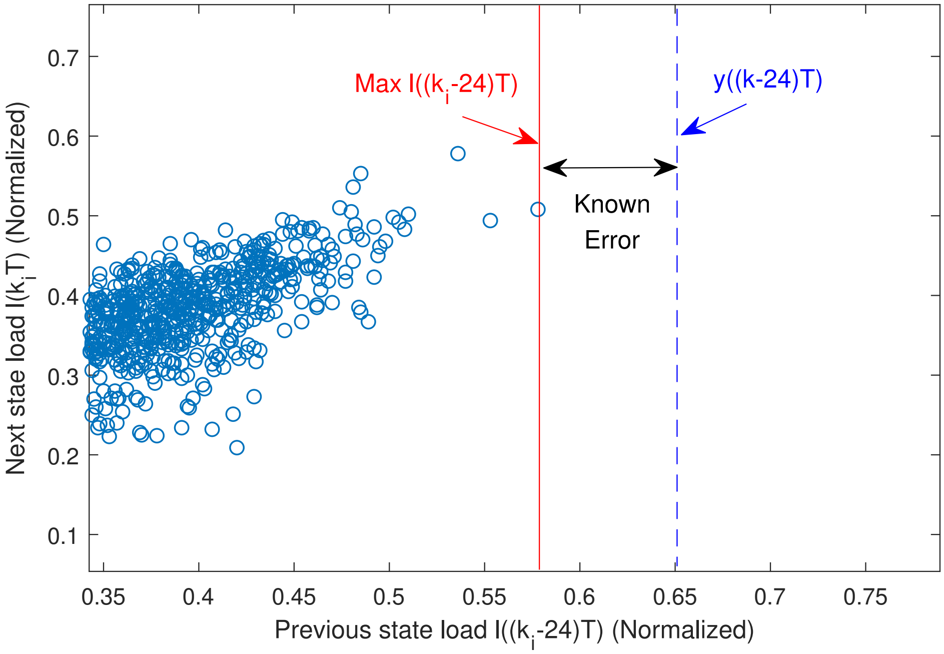
3.3.2 Compensation

Another problem occurs when the reference load deviates from the training set range. The two cases of that condition are shown in Fig. 3.7. When the reference load $y((k-24)T)$ is smaller than the minimum of the training data or bigger than the maximum, the observed points satisfying $l((k_i - 24)T) = y((k-24)T)$ cannot be found, which are needed to set up the next state vector $\vec{y}$ and condition matrix C of Eq. (2.3).

Being out of range means that an unprecedented load has emerged that has never been learned in the past. Since we have no choice but to use the training set, the reference load must be treated as the minimum value (or maximum value), even if it is smaller than the minimum. One benefit is that we already know the error due to this forced approximation. This fact presents us with the main idea to solve this problem. Once the estimated load is obtained using this forced approximation, we can compensate for the intentional error from what we already know. Let



(a) Case of the reference load $y((k-24)T)$ being smaller than the minimum of the training set.



(b) Case of the reference load $y((k-24)T)$ being bigger than the maximum of the training set.

Figure 3.7: Examples that show the reference load out of the training set range.

Δ^- and Δ^+ denote these intentional errors, then the estimated load $\hat{y}(kT)$ in Eq. (3.9) should be adjusted as follows.

- Case where $y((k-24)T) < \min\{l((k_i-24)T)\}$

$$\begin{aligned} &\Delta^- \triangleq \min\{l((k_i-24)T)\} - y((k-24)T) \\ &\hat{y}(kT) = \vec{c}_k \cdot \vec{x}_k^* - \Delta^- \end{aligned} \tag{3.12}$$

- Case where $y((k-24)T) > \max\{l((k_i-24)T)\}$

$$\begin{aligned} &\Delta^+ \triangleq y((k-24)T) - \max\{l((k_i-24)T)\} \\ &\hat{y}(kT) = \vec{c}_k \cdot \vec{x}_k^* + \Delta^+ \end{aligned} \tag{3.13}$$

The symbol Δ^- represents an intentional error to compensate for when the reference load is less than the minimum of training data set, and similarly, Δ^+ is also an intentional error to compensate for when the reference load is greater than the maximum.

3.3.3 Flow Chart

Fig. 3.8 shows the flow chart of the PMLR algorithm to solve the application problems. Once the reference load is set, we can find the past observed points that are the same or close to the reference from the training set. If we can search for the sufficient number of observed points ($m > d$), the optimized unknown vector $\vec{x}_k^*$ can be calculated directly. Otherwise ($m < d$), we need to check whether the reference load $y((k-24)T)$ is within the training set range. When the reference load is within the range but the number of searched points m is small, the searching interval ε_k should be increased until m becomes sufficient. When the reference load is out of the training set range, the prediction is performed by compensating for the intentional errors from what we already know.

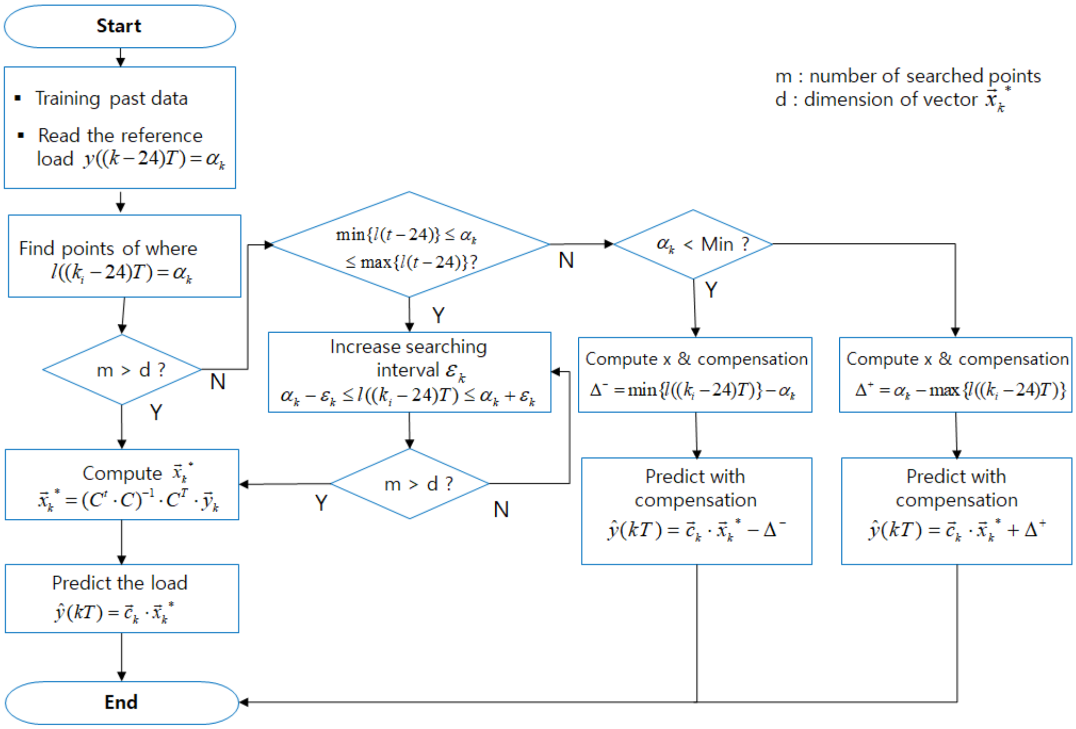

Start
Training past data
Read the reference load $y((k-24)T)=\alpha_k$
Find points of where $l((k_i-24)T)=\alpha_k$
m > d ?
N
Y
$\min\{l(t-24)\}\le\alpha_k$ $\le\max\{l(t-24)\}?$
N
Y
Increase searching interval ε_k
$\alpha_k-\varepsilon_k\le l((k_i-24)T)\le\alpha_k+\varepsilon_k$
m > d ?
Y
N
Compute $\vec{x}_k^*$
$\vec{x}_k^*=(C^t\cdot C)^{-1}\cdot C^T\cdot\vec{y}_k$
Predict the load
$\hat{y}(kT)=\vec{c}_k\cdot\vec{x}_k^*$
End
α_k < Min ?
Y
Compute x & compensation
$\Delta^-=\min\{l((k_i-24)T)\}-\alpha_k$
Compute x & compensation
$\Delta^+=\alpha_k-\max\{l((k_i-24)T)\}$
Predict with compensation
$\hat{y}(kT)=\vec{c}_k\cdot\vec{x}_k^*-\Delta^-$
Predict with compensation
$\hat{y}(kT)=\vec{c}_k\cdot\vec{x}_k^*+\Delta^+$
m : number of searched points
d : dimension of vector $\vec{x}_k^*$

Figure 3.8: Flow chart of the PMLR algorithm.

3.4 Application of Weighted Least Squares

The previously outlined GMLR and PMLR algorithms solve the OLS problem based on the assumption that the observed load values have the same error variance. However, in most cases, the error variance might be different on each observation. Considering this fact, we can establish the weight for each observation, which can be regarded as degree of belief. The prediction performance may vary depending on how the weight is defined, as described in Section (2.2). In this section, we will introduce several methods to estimate the weight matrix, W, and we will compare the performances of GMLR and PMLR for each method in the next chapter.

3.4.1 Sample Variance of Ordinary Least Squares

The most commonly used method for estimating the weight is to use an error variance estimator. If we run the OLS first, the squared residuals between the observed values and the estimated values from OLS can be achieved. White [Whi80] suggests this method in his paper (1980), estimating each weight to be a reciprocal of the squared residuals. Let e_i denote i-th error defined in Eq. (2.1) and Eq. (2.2). Then, the i-th weight, w_i, can be defined as follows.

$$w_i = \frac{1}{{e_i}^2} = \frac{1}{(y_i - \sum_j c_{ij}{x_j}^*)^2}, \quad i \in [1,\, m]. \tag{3.14}$$

If we have m observed values, the weight w_i in Eq. (3.14) is the i-th observation weight of the total m observations, where y_i is the i-th observed load (actual value) and $\sum_j c_{ij}{x_j}^*$ is the i-th estimated value by the OLS method. The w_i is a reciprocal of the squared error or variance between the past actual load and the estimated load. This weight can be interpreted as an expected precision belief for each observation. When we have no information about the observations, this suggestion may be our best choice.

3.4.2 Temperature Difference

The m observed points of the GMLR and PMLR methods are assumed to have a similar property with the target load of what we want to predict. If the error variance of these observed data depends on a certain variable such as temperature, the weight can be described as a function of that variable. Fig. 3.1 shows that the load and temperature are highly correlated each other, in which this fact gives us an idea that the temperature difference between the target day and the past observed day might indicate the weight of each observation. We can expect that as the temperature of the past observed day is close to that of the target day, the prediction would be precise. Therefore, this temperature difference can be used to establish the weight, i.e. the belief degree for each observed data. Let h_i denote the i-th observed temperature among the total m observations, h_T denote the temperature of the target time, and ξ denote a small positive number to prevent the denominator equaling zero. The i-th weight, w_i, will be

$$w_i = \frac{1}{|h_i - h_T| + \xi}, \ i \in [1,\ m]. \tag{3.15}$$

If the temperature difference is large, since the estimated data is expected to have a large error, we trust them less. On the contrary, if the temperature difference is small, we can trust the observed data more.

3.4.3 Realization Probability

According to the central limit theorem (CLT), the probability distribution of a random variable converges to the normal distribution as we observe them many times independently, even though the actual observed random variable is not normally distributed. Applying this theorem to the observed data of this paper, we can regard the observed data as a random variable. Even if they are not normally distributed, as we increase the number of observations, their probability density function (PDF) would converge on the normal distribution. Since we can calculate the mean and variance of the chosen m observed data, we can find all the information about their PDF by CLT. We can calculate the probability of each realized observation, implying that the probability of the near-average values is high. If we apply this hypothesis to establish the observation weight for the chosen m observed data, we can define the weight using the PDF of the normal distribution. Let y_i denote i-th observed load, μ denote the sample mean of y, and σ denote the variance of y. The i-th weight, w_i, is

$$w_i = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(y_i - \mu)^2}{2\sigma^2} \right), \ \ i \in [1,\ m]. \tag{3.16}$$

This method presents a trust degree for each observation, which allocates a larger weight for more frequently appearing data. Conversely, the observed data that do not appear often will be given a smaller weight.

3.5 Grouping Methods of the Training Set

Unlike the GMLR prediction, which has a short (one month) and fixed observation period, the PMLR prediction may have various performances depending on how we handle the training set. In our study, we will use the data from 2015.01 $\sim$ 2016.06 (18 months) as the training term and the data from 2016.07 $\sim$ 2016.12 (6 months) as the verification term. If the training set can be divided and chosen according to the characteristic of the target day, the prediction performance might be improved. In other words, as the correlation between the previous state (reference load) and the next state is stronger, the prediction performance improves. Based on this hypothesis, we divide the training set with a total of 18 months by the following five methods and compare the prediction performance according to each method.

The first and second grouping methods consider the seasonal characteristics of the target day and the remaining three grouping methods apply *k*-means clustering. The first grouping method (G1) is a seasonal method, which divides the 18 months of data into 3 parts (summer, winter, and the rest) in a mutually exclusive manner. Spring and fall data are regarded as having a similar property. The G1 has the advantage that no data is discarded because it uses all 18 months of data. The second grouping method (G2) is also a seasonal method, which uses 4 months of data closest to the forecasting target month. For example, the data of June $\sim$ September (4 months) are used to predict the load of July and August. With this method, we only use 8 months of data among a total of 18 months and the remaining 10 months of data are not used. Thus, the G2 has the drawback of information loss. Table 3.1 represents the seasonal grouping of the training set using the G1 and G2 methods.

The 3rd, 4th, and 5th grouping methods divide the training set with 18 months into several parts using the *k*-means clustering algorithm. Since the *k*-means clustering algorithm

Table 3.1: Seasonal grouping of the training set with the G1 and G2 methods.

Prediction target month	G1 training set	G2 training set
2016. 7 $\sim$ 8	2015. 5 $\sim$ 8, 2016. 5 $\sim$ 6	2015. 6 $\sim$ 9
2016. 9 $\sim$ 10	2015. 3 $\sim$ 4, 2015. 9 $\sim$ 10, 2016. 3 $\sim$ 4	2015. 8 $\sim$ 11
2016. 11 $\sim$ 12	2015. 1 $\sim$ 2, 2015. 11 $\sim$ 2016. 2	2015. 10 $\sim$ 2016. 1

Table 3.2: Different options for *k*-means clustering of the training set with the G3, G4, and G5 methods.

Option	G3	G4	G5
Number of clusters (*k*)	6	3	9
Option for cluster selection	singular	singular	multiple

divides the data by the index we set, it is imperative to set the index that best represents the characteristics of each data. The clustering results may vary depending on how the index was set, which significantly impacts the prediction performance. We establish the *k*-means clustering index as [Month, Load, Temperature] to reflect the seasonal property of each date and the related tendency of electric load and temperature, which we use in the load modeling equation. The reference index of the prediction target month is set to be an average index of the same month of the previous year. The load of the target month can be predicted by selecting the cluster that has the most resemblance to the reference index. The three methods of *k*-means clustering are determined by how many clusters we divide the training set into and how many clusters we select per target month. The 3rd grouping method (G3) divides the training set into 6 clusters and chooses a singular cluster for each target month, while the 4th grouping method (G4) divides the training set into 3 clusters and chooses a singular cluster for each target month. Lastly, the 5th grouping method (G5) divides the training set into 9 clusters and chooses multiple clusters for each target month. Since the optimized clustering cannot be known, we tried to find the best clustering options for our training set by using combinations of *k* and cluster selection manners. Table 3.2 summarizes these different options for *k*-means clustering of the training set with the G3, G4, and G5 methods.

Chapter 4

Case Study

In this chapter we will show the results of two case studies. For the first case study, we will use real electric data spanning 2 years from 2015 to 2016 in Austin, Texas, USA in order to compare the performances of GMLR and PMLR. We will look into the inherent limitation of the GMLR method and create a comparative study of the PMLR method as a replacement for the GMLR under certain conditions. We will also apply weighted least squares (WLS) for both methods with 3 different types for setting the weight and implement the 5 different grouping methods, which were introduced in Chapter 3. Another case study will be conducted to verify our hypothesis using the demand forecast data in Jeju Island, South Korea. This additional experiment will prove that our suggestion can be generally accepted.

4.1 Experiment Setup

4.1.1 Data Preprocessing

In our study, we use hourly electric load data spanning 2 years from 2015 to 2016 in Austin, Texas, USA, which is provided by Pecanstreet.org. The residences with photovoltaic cell are excluded from the analysis to avoid substantial data variations. The regional weather

Table 4.1: Temperature on 2015-01-01 19:00 $\sim$ 20:00 in Austin, Texas, USA.

Time	**Temperature ($^\circ F$)**
2015-01-01 19:05	37
2015-01-01 19:35	38
2015-01-01 19:55	38

information is obtained from the National Oceanic and Atmospheric Administration (NOAA) Bergstrom station in Austin. The hourly wet bulb temperature from the weather information is implemented for the prediction. We use the data from 2015.01 $\sim$ 2016.06 (18 months) as the training term and the data from 2016.07 $\sim$ 2016.12 (6 months) as the verification term.

Both the loads from Pecanstreet and the temperatures from NOAA have some data loss. The lost data is interpolated into the average of the before and after values of the lost moment. Let $D(kT)$ be the lost data at a specific time kT for both the load and temperature. Then, $D(kT)$ is interpolated as follows.

$$D(kT) = \frac{D((k-1)T) + D((k+1)T)}{2} \tag{4.1}$$

The temperature data from NOAA were not hourly-based. Instead, there were two or three data per hour. To convert it to hourly data, the earliest data was picked as the corresponding time data on every hour. For example, if there are three data on 2015-01-01 19:00 $\sim$ 20:00 as shown in Table 4.1, then we select $37^\circ F$ as the hourly temperature data at 19:00. Since the temperature change during the one hour is trivial, there is no harm in picking one of them as the representative value.

We utilize data consisting of about 300 residences provided by Pecanstreet in this paper, but not all residences have complete data within the 2 years from 2015 to 2016 as shown in Table 4.2. Complete data spanning 2 years has 17,520 hourly data, as is the case for residence ID numbers 59 and 86. We will use an aggregated load data of some residences, which have complete data for the 2 years, not an individual load. The load prediction of an individual residence requires

Table 4.2: Number of hourly data of each residence ID in Austin, Texas, USA.

Residence ID	22	59	68	86	$\cdots$	9981
Number of Data	4,039	17,520	5,522	17,520	$\cdots$	4,012

much more sophisticated and complicated algorithms due to its load pattern holding random properties with many variables to be considered according to the characteristics of each residence, such as the type of building, income level, number of family members and so on. The aggregated load of a large scale at a national or urban level has seasonal characteristics and repeated patterns every 24 hours, but an individual load does not. In fact, the individual load patterns of the 64 residences selected for this paper are completely different, making it difficult to see repeated patterns every 24 hours. However, the aggregated data of multiple residences offset the random properties and have a 24-hour periodic pattern. Silvar [SIK14] showed in his paper (2014) that the prediction for grouping customers with similar properties would be more accurate than an individual customer. In this paper, we focus on predicting the load at the small community level in order to provide information for efficiently managing the power system rather than predicting the random and irregular load of an individual residence. To do this, we choose 64 residences who have complete load data within the 2 years and employ their average as the load data for our experiment.

When the load data and the temperature data of different units are used together as in Eq. (3.1), a large prediction error may occur. To avoid this, we utilize normalization. Since the ranges of load and temperature are different, rescaling is performed as follows to convert the both ranges to [0, 1]. Let D denote the original value of the load and temperature data and D_{nor} denote the normalized value. Then, the D_{nor} is given as:

$$D_{nor} = \frac{D - \min(D)}{\max(D) - \min(D)}. \tag{4.2}$$

We will use this normalized value for the electric load and temperature data in this paper.

4.1.2 Performance Metrics

The forecasting performance is measured by the RMAPE (Relative Mean Absolute Percentage Error), which is calculated on a daily basis. Rahman [RH93] employed RMAPE in order to reduce the near-zero distortion instead of the commonly used MAPE. Unlike RMAPE, MAPE is calculated on a hourly basis. Let $\hat{y}(kT)$ denote the estimated load, $y(kT)$ denote the actual load at time kT, and y_{pod} denote the peak load of the day. Then, the RMAPE of the day and MAPE will be given by

$$RMAPE = \frac{1}{24}\sum_{k=1}^{24}\frac{|\hat{y}(kT)-y(kT)|}{y_{pod}} \;\; (\%) \tag{4.3}$$

and

$$MAPE = \frac{|\hat{y}(kT)-y(kT)|}{y(kT)} \;\; (\%). \tag{4.4}$$

MAPE in Eq. (4.4) has a drawback in that it becomes infinite when the denominator is zero. If we use normalized data, the denominator of MAPE often becomes a zero value and this leads an unexpected error. We can address this problem by employing RMAPE, which applies the largest load of the day to the denominator as in Eq. (4.3). Since the peak load of the day cannot be zero, RMAPE cannot be infinity.

4.2 Experiment Results

4.2.1 Results of the OLS Method for the GMLR and PMLR

We conduct the first case study with real data of 2 years in Austin, Texas, USA. The data from 2015.01 $\sim$ 2016.06 (18 months) is used as the training term and the data of 2016.07 $\sim$ 2016.12 (6 months) as the verification term. Since we have 7 unknowns for the unknown vector

Table 4.3: Performance of GMLR depending on the number of equations in Austin, Texas, USA.

Num. of Eq.	10	20	30	40	50	60	70
GMLR RMAPE (%)	17.97	10.78	9.80	9.89	10.08	10.34	10.78

Table 4.4: Performance of PMLR depending on the number of equations in Austin, Texas, USA.

Num. of Eq.	30	50	80	100	200	300	400
PMLR RMAPE (%)	11.40	10.94	10.65	10.59	10.40	10.34	10.32

$\vec{x}$ of both the GMLR in Eq. (3.2) and PMLR in Eq. (3.6), we need at least 7 equations to solve an OLS problem. The number of equations refers to the number of chosen observed data from the training set. Generally, this number of equations (or the number of observed data) affects the performance of the MLR prediction. Table 4.3 and Table 4.4 show the average RMAPE for 6 months of GMLR and PMLR depending on the number of equations. The performance of GMLR in Table 4.3 is the best, 9.80%, when the number of equations is near 30. On the other hand, the performance of PMLR in Table 4.4 improves as the number of equations increases and converges to the best value, 10.34%, when the number of equations is near 300. Even though the prediction performance of the GMLR and PMLR presents a different tendency depending on the number of equations, we set the number of equations to 30 and regard it as a constant for both methods considering their equality. We treat all of predictions with the 30 chosen observed data and compare the performance of each prediction method.

Now we start with the OLS results of the GMLR prediction. Fig. 4.1 shows the daily RMAPE of the OLS-GMLR for 6 months (2016.6 $\sim$ 2016.12). The performance tends worsen as it becomes winter. The monthly RMAPE box plot shown in Fig. 4.2a illustrates this tendency more explicitly. On each box, the bottom and top lines of the box indicate the 25th and 75th percentiles respectively, and the central mark indicates the median. The outliers are plotted individually using the '+' symbol. The prediction performance of November and December is

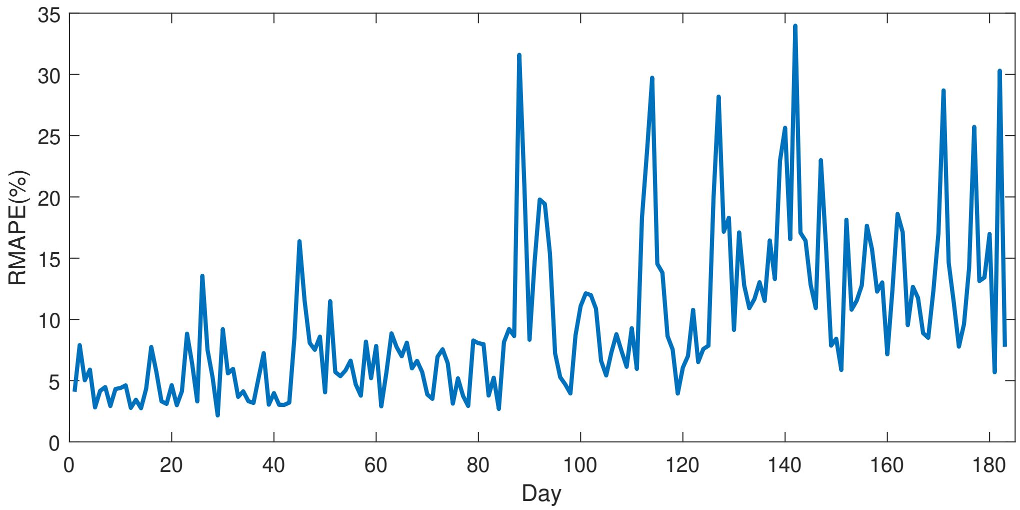
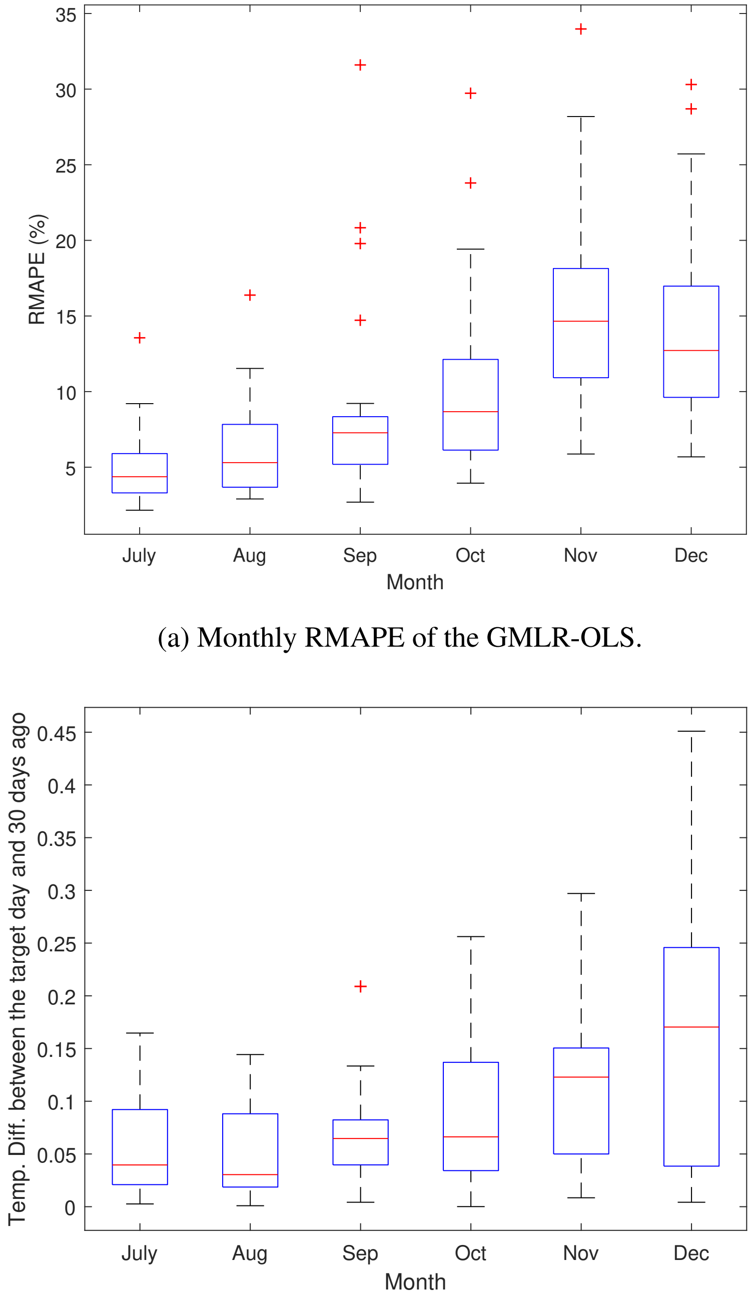


Figure 4.1: Daily RMAPE of GMLR for 6 months (2016.6 ∼ 2016.12) in Austin, Texas, USA.

worse compared to that of the previous months. Since we choose the observed data on a time base in the GMLR prediction, the prediction performance depends on the time effect. Therefore in this GMLR, the error may increase when a specific variable such as temperature changes suddenly during the observation period. Fig. 4.2b shows the monthly temperature difference distribution between the target day and 30 days prior. Fig. 4.2b also indicates that the temperature difference of November and December is high, which might affect the GMLR performance. The Pearson correlation coefficient of the GMLR RMAPE and the temperature difference is 0.5677, which indicate they are correlated each other. Since this GMLR performance is correlated with the temperature difference, we can imagine that if there is a new MLR method that is independent from the temperature difference, we can improve the poor performance of the GMLR with a new MLR.

For this reason, we propose a new MLR that selects the observed data on a specific past load basis, not time basis. The PMLR does not have the observation period, but instead, predicts the target load with the reference load by learning the transition tendency in the training set. Since this PMLR selects the observed data based on a specific value, the prediction performance is independent from the temperature difference. The Pearson correlation coefficient of the PMLR



(a) Monthly RMAPE of the GMLR-OLS.

(b) Monthly temperature difference.

Figure 4.2: Monthly RMAPE and temperature differences in Austin, Texas, USA.

Table 4.5: Monthly average RMAPE of the GMLR and PMLR predictions in Austin, Texas, USA.

Month	7	8	9	10	11	12	Average
GMLR RMAPE (%)	5.06	6.04	8.35	10.39	15.38	13.94	9.80
PMLR RMAPE (%)	6.21	8.93	9.50	13.64	13.68	16.67	11.41

RMAPE and the temperature difference is 0.4185, which indicates that they are less correlated each other. Table 4.5 shows the monthly average RMAPE of the GMLR and PMLR predictions. In November, the PMLR prediction performance is better than the GMLR, which is supposedly due to a temperature independency. When the prediction performance of the GMLR degrades due to rapid changes in temperature, the PMLR method can replace the GMLR and improve the prediction performance. To summarize, Table 4.5 shows that the GMLR prediction performance is better than the PMLR for the 6 months overall, but the PMLR prediction performance can be beneficial in special cases such as November because of its temperature independency.

4.2.2 Results of the WLS Method for the GMLR and PMLR

In the previous section, we have seen the performance results of the GMLR and PMLR predictions in solving the OLS problem. This OLS problem assumes that all of the chosen observed data have the same error variance. However, in practice, the error variance might be different on each observation. Applying this hypothesis, we have conducted several experiments for the GMLR and PMLR solving the WLS problem. In this section, we try to compare the prediction performances of the GMLR and PMLR by implementing the three methods of establishing the observation weight described in Chapter 3.4. The prediction performance can be improved depending on how the weights are designed.

The design of the weights on each observation is closely related to how much we trust the chosen observed data. The first method (M1) described in subsection 3.4.1 utilizes the sample variance of the OLS estimation. We trust the observed data as much as the reciprocal of

Table 4.6: Average RMAPE for 6 months of the GMLR and PMLR predictions with 3 WLR methods in Austin, Texas, USA.

Application Method	OLS	WLS-M1	WLS-M2	WLS-M3
GMLR RMAPE (%)	9.80	9.82	10.10	9.71
PMLR RMAPE (%)	11.41	11.45	11.58	10.96

Table 4.7: Average RMAPE for November of the GMLR and PMLR predictions with 3 WLR methods in Austin, Texas, USA.

Application Method	OLS	WLS-M1	WLS-M2	WLS-M3
GMLR RMAPE (%)	15.38	15.42	15.77	14.95
PMLR RMAPE (%)	13.68	13.72	15.15	12.63

their squared residuals. The second method (M2) in subsection 3.4.2 exploits the temperature difference between the target time and the observed time. We trust the observed data more if the temperature difference is small. The last method (M3) in subsection 3.4.3 applies the realized likelihood of the observed data. We trust the observed data depending on how often they appear. Table 4.6 shows the average RMAPE for 6 months of the GMLR and PMLR predictions with 3 WLR methods. The GMLR performance with WLS is similar to the one with OLS, while the PMLR performance is improved by applying the WLS-M3. The improvement of prediction performance is not clearly seen with the overall average RMAPE for 6 months. But if we focus on the performance in November when the GMLR performs poorly, we can clearly see improvement in the performance of the PMLR prediction when applying the WLS-M3 as shown in Table 4.7. While the GMLR prediction performance with OLS for November is poor with 15.38%, the PMLR prediction performance with WLS-M3 improves with 12.63%, which means the performance improvement is as much as 2.75%.

We can see this performance improvement explicitly with the comparison of the load patterns between the real and the estimated. Fig. 4.3 shows the comparison of the actual electric load (Blue line with no mark) and two estimated loads using the GMLR-OLS (Dashed red line with cross mark) and the PMLR-WLS-M3 (Dashed black line with diamond mark) in November.

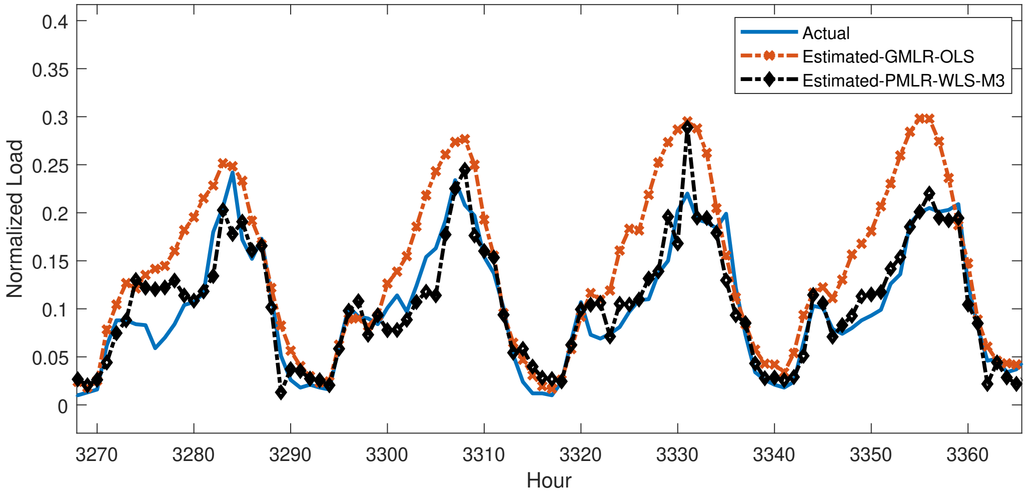


Figure 4.3: Comparison of the actual electric load (Blue line with no mark) and two estimated loads using the GMLR-OLS (Dashed red line with cross mark) and PMLR-WLS-M3 (Dashed black line with diamond mark) during November in Austin, Texas, USA.

As we can see, the dashed red line with cross mark (the GMLR-OLS prediction) could not follow the blue line (the real electirc load) closely, which indicates poor prediction performance. On the other hand, the dashed black line with diamond mark (the PMLR-WLS-M3 prediction) is close to the blue line, indicating good predicion performance. To summarize, when the GMLR prediction performance degrades with rapid weather change, it is preferable to use the PMLR method rather than simply using the GMLR. Additionally, the prediction performance can be further improved by the additional application of the proper WLS technique.

4.2.3 Results of Grouping the Training Set for the PMLR

As introduced in section 3.5, the better prediction performance can be expected if we forecast the load with a selected partial training set that has similar data properties with the target time. The most intuitive and easiest way of grouping the whole training set is to utilize the date of data that represents the seasonal property of data. If we predict a summer load only with a summer training set or predict a winter load with a winter training set, the prediction becomes

Table 4.8: Monthly RMAPE of the PMLR-OLS with the G1 and G2 grouping methods for Austin, Texas, USA.

Grouping method	7	8	9	10	11	12	Average
No grouping	6.21	8.93	9.50	13.64	13.68	16.67	11.41
G1 (seasonal)	6.28	8.22	9.17	12.93	14.29	14.74	10.94
G2 (seasonal)	5.91	8.47	10.88	17.45	15.17	15.89	12.29

more accurate because the variance of the selected training set is reduced. Table 4.8 shows the monthly RMAPE of the PMLR-OLS with G1 and G2 grouping method for Austin, Texas, USA. The prediction performance of the original PMLR-OLS with no grouping is compared to the results of the G1 and G2 grouping methods. The G1 grouping method improved overall performance for 6 months, but the performance during November is rather degraded. But the G2 grouping method not only decreases the overall performance for 6 months, but also worsens the performance in September $\sim$ November. The reason for the lower performances of the G1 and G2 groupings seems to be a date-dependent contrived grouping manner. Since Austin has vague seasonal boundaries, it is not easy to clearly distinguish between summer and autumn or autumn and winter by date alone. Even though we want to collect solely summer data to forecast a summer load, some spring or fall or even winter data might have been included with the chosen summer training set and contributed to the error.

On the other hand, the k-means algorithm divides the training set into k clusters by the new index we set, not by the date, and so we expect more reasonable grouping results and improved prediction performances than the G1 and G2 groupings. Table 4.9 shows the monthly RMAPE of the PMLR-OLS with the G3, G4, and G5 grouping methods for Austin, Texas, USA. The November performance of G3 is better than that of G1, but the overall performance of G3, which is much better than that of G2, is similar to that of G1, which does not use k-means algorithm. While the seasonal G1 grouping utilizes the whole training set of 18 months, the k-means G3 grouping does not exploit the whole training set, meaning G3 has some data loss that leads to a limitation in improving the prediction performance. In actuality, the G3 grouping method utilizes

Table 4.9: Monthly RMAPE of the PMLR-OLS with the G3, G4, and G5 grouping methods for Austin, Texas, USA.

Grouping method	7	8	9	10	11	12	Average
G3 (*k*-means)	6.14	8.37	9.54	13.54	12.94	15.29	10.97
G4 (*k*-means)	6.14	8.72	9.69	14.16	13.12	15.59	11.24
G5 (*k*-means)	6.06	8.40	9.29	13.32	13.01	15.00	10.85

only 3 clusters among 6 to forecast the load (Usage rate: 35.1%), so we tried to find another method that minimizes discarded data. The clusters number of the G4 *k*-means grouping is set to reduce from 6 to 3. We have expected that all of 3 clusters of G4 would be used for prediction, since the whole training set could be categorized into 3 seasonal parts such as summer, winter, and fall (or spring). However, unlike our expectation, the G4 method also does not use all clusters but uses only 2 clusters among 3 (Usage rate: 51.9%). Although G4 shows a higher data usage than G3, it still suffers from data loss. Additionally, its prediction performance is even worse than that of G3 or G1, caused by containing a lot more noise in the chosen cluster. We tried the next 5th grouping method (G5) in order to minimize discarding training data and find out the optimized clustering for the target month. If the *k*-means clustering inevitably produces some discarded data and the optimized clustering is not known, it may be a good strategy to divide the training set into many clusters as possible and choose multiple clusters for the target month. G5 has used 5 clusters among 9 (Usage rate: 46.1%), in which its data usage is lower than that of G4, but the multiple choices of small clusters eliminate the noise effectively and eventually improve the prediction performance. The best performance of the G5 grouping method in Table 4.8 and 4.9 demonstrates our argument.

If we go deeper, these training set grouping techniques and the previous WLS methods can be applied at the same time since they are independent each other. Combining the *k*-means G5 grouping, which is known to be the best in this section, and the WLS-M3 method, which was identified to be optimal in the previous section 4.2.2, can further improve our PMLR prediction performance. Table 4.10 shows the monthly RMAPE of the PMLR-WLS-M3 with no grouping

Table 4.10: Monthly RMAPE of the PMLR-WLS-M3 with no grouping and PMLR-WLS-M3 with G5 grouping in Austin, Texas, USA.

Method	7	8	9	10	11	12	Average
PMLR-WLS-M3 with no grouping	6.01	8.83	9.37	13.60	12.63	15.54	10.96
PMLR-WLS-M3 with G5 grouping	5.86	8.30	9.21	13.23	12.25	14.61	10.58

and PMLR-WLS-M3 with G5 grouping in Austin, Texas, USA. The PMLR-WLS-M3 with G5 grouping method shows better performance in all months than the PMLR-WLS-M3 with no grouping method, making it our best prediction performance in this paper.

4.3 Expansion of the Experiment

In this section, we conduct another case study to verify if our argument can be generally accepted. We will use an electric demand forecast data set of Jeju, South Korea. A synthetic electric load data set could be used to demonstrate our argument, but it is very difficult to create one that is close to reality, and it also may be unreliable. Since this demand forecast data set is not true measured data, it might be unsatisfactory for use in our study, but it is more realistic to use this data set than the factitious synthetic data set.

The electric demand forecast data set of Jeju is provided by the Korea Power eXchange (KPX) and we will use 2 years worth of data from 2015 to 2016. Simliar to the previous case study in Austin, Texas, USA, we will use the data from 2015.1 $\sim$ 2016.6 (18 months) as the training set and verify the prediction with the data from 2016.7 $\sim$ 2017.12 (6 months). The weather information of Jeju is provided by the Korea Meteorological Administration (KMA), in which we will use hourly temperature data for 2 years. Since both the electric load data and weather data are on an hourly basis with no missing data, we do not need to perform any preprocessing or interpolation that was required in the previous case study. However, as the units are different, we need to perform normalization by converting the value to [0, 1] using Eq. (4.2).

Table 4.11: Monthly average RMAPE of the GMLR and PMLR with the OLS and 3 WLS methods in Jeju, South Korea.

Method	7	8	9	10	11	12	Average
GMLR-OLS	8.74	6.79	9.25	9.37	7.58	8.33	8.31
GMLR-WLS-M1	8.77	6.79	9.25	9.38	7.59	8.35	8.32
GMLR-WLS-M2	8.75	6.86	9.37	9.40	7.62	8.39	8.36
GMLR-WLS-M3	8.76	6.81	9.32	9.22	7.63	8.41	8.32
PMLR-OLS	7.81	7.93	8.14	8.30	6.12	9.00	7.87
PMLR-WLS-M1	7.77	7.83	8.14	8.27	6.11	8.93	7.83
PMLR-WLS-M2	8.28	8.17	8.25	7.88	6.63	9.16	8.06
PMLR-WLS-M3	7.78	7.62	8.04	7.97	6.04	8.99	7.72

With these hourly data of load and temperature normalized, we execute the same experiments for the GMLR and PMLR predictions and solving the OLS or WLS problems. Table 4.11 shows the monthly average RMAPE of the GMLR and PMLR with the OLS and 3 WLS methods in Jeju. These results are somewhat different from the previous case study in Austin. In this case, the PMLR prediction performance for 6 months overall is better than that of the GMLR. Unlike the previous results in Austin, the GMLR RMAPE is less correlated with the temperature difference between the target day and 30 days ago. This low correlation is supposedly due to there being less sudden temperature changes due to seasonal effects in Jeju.

However, on the whole, Table 4.11 shows similar results with the previous case study in Austin. The GMLR prediction performance shows no improvement, even if we apply the 3 WLS methods. But the PMLR prediction performance can be improved, if we apply a proper WLS method. With the exception in August and December, the PMLR prediction performance is better than that of the GMLR. We can improve the PMLR prediction performance further by applying the WLS-M1 in July, WLS-M3 in September, WLS-M2 in October, and WLS-M3 in November. The WLS-M3 method shows the best performance overall, while in some cases, the WLS-M1 or M2 method performs better. If we focus on the PMLR prediction performance in November when the difference between the GMLR and PMLR performances is the largest, we can improve the performance with RMAPE 1.54% by applying the PMLR-WLS-M3 prediction (RMAPE

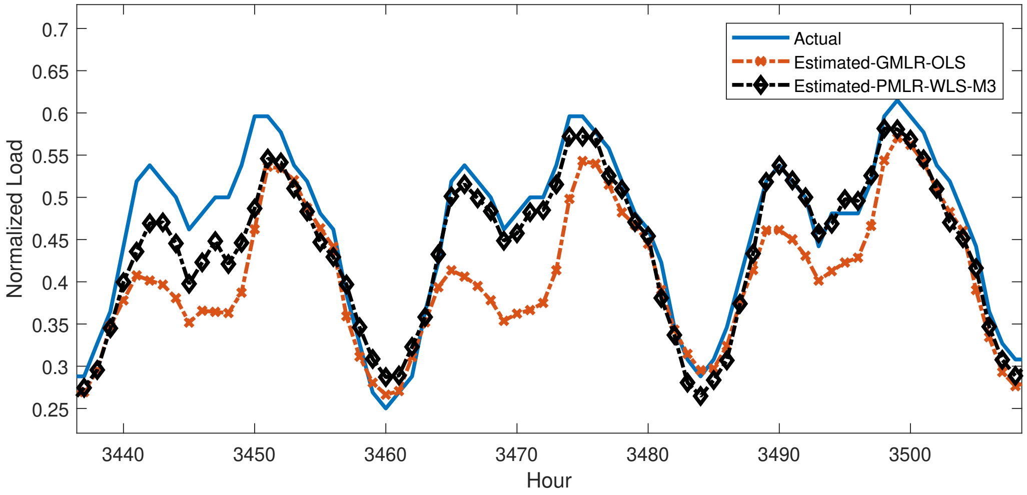


Figure 4.4: Comparison of the actual electric load (Blue line with no mark) and two estimated loads using the GMLR-OLS (Dashed red line with cross mark) and PMLR-WLS-M3 (Dashed black line with diamond mark) of November in Jeju, South Korea.

6.04%) instead of applying the GMLR-OLS prediction (RMAPE 7.58%). We can observe this performance improvement more clearly with a comparison of the load pattern. Fig. 4.4 shows the comparison of the actual electric load (Blue line with no mark) and two estimated loads using the GMLR-OLS (Dashed red line with cross mark) and PMLR-WLS-M3 (Dashed black line with diamond mark) during November in Jeju, South Korea. Like the previous case study in Austin, the load pattern with the PMLR-WLS-M3 prediction (Dashed black line with diamond mark) is closer to the actual load pattern than the GMLR-OLS prediction pattern (Dashed red line with cross mark). Fig 4.4 demonstrates that our argument about WLS method can be also applied to the second case study in Jeju.

However, an additional experiment involving *k*-means clustering of the Jeju data set shows rather different results from the Austin data set. As we have seen from the previous sections, the PMLR prediction performance can be improved the most by applying the WLS-M3 and G5 grouping methods. Therefore, we implemented the same methods to the Jeju data set, in which the results are shown in Table 4.12. Contrary to our expectation, the prediction performance of the Jeju data set was deteriorated by grouping the training set. The original method of using the whole

Table 4.12: Monthly RMAPE of the PMLR prediction with the WLS-M3 and G5 methods in Jeju, South Korea.

Method	7	8	9	10	11	12	Average
PMLR-OLS with no grouping	7.81	7.93	8.14	8.30	6.12	9.00	7.87
PMLR-OLS with G5 grouping	8.18	9.63	8.40	8.00	6.99	9.55	8.46
PMLR-WLS-M3 with no grouping	7.78	7.62	8.04	7.97	6.04	8.99	7.72
PMLR-WLS-M3 with G5 grouping	8.11	9.03	8.10	7.72	6.71	9.51	8.20

training set for prediction shows better performance than the grouping method of the training set. The reason is presumed to be that the climate properties and load usage pattern of Jeju are different from those of Austin. When the Jeju training set is grouped in the same way as the training set in Austin and ignoring the fact that they have different climate properties, the variance in the correlation graph of the previous state and the next state is increased in a certain section for the Jeju data set. This increase in variance degraded the prediction performance in a certain section. It was a miscalculation to assume that the same approach would apply appropriately to the Jeju data set, even though the climate characteristics of Jeju and Austin are quite different. We need a completely different approach to improve the prediction performance for Jeju such that the humidity, wind speed, and other variables are added to the clustering index or the clustering number *k* is modified to be more than 9.

Through the Jeju case studies, we have shown that the prediction performance could be improved by replacing the GMLR prediction with the PMLR prediction under a certain condition and further enhanced by applying the proper WLS method according to the case. However, the PMLR prediction performance with *k*-means clustering can vary depending on how we set the clustering index or other options.

Chapter 5

Conclusion

In this paper, we formulate the present electric load with an MLR model, which is an inner product of the condition vector and the unknown coefficient vector. The condition vector is given by the past load and temperature data and the unknown coefficient vector is calculated by solving the least squares problem. If we properly collect some observed data, we can calculate the optimized coefficient vector that minimizes the sum of squared errors. With this optimization, we can predict the target load.

Since the GMLR prediction selects the observed data on a time basis, its performance can be degraded due to the time or seasonal effects. For this reason, we proposed a new method of MLR (PMLR) which selects the observed data on a specific past load basis. Although there are two problems to apply the PMLR, we could address them with some clever techniques. If we have insufficient observed points, we can obtain more of them through approximately adaptive searching. When the reference load is out of the training data set, the prediction error can be reduced by compensating the forced error, which we already know.

For further prediction improvement, we tried to set the weight of each observation and calculate the optimized coefficient vector by solving the WLS problem. The prediction performance depends on how the weight is modeled and we introduce several methods as follows.

In the first method, we exploit the sample variance with the OLS estimated value. In the second, the weight is defined by using the temperature difference between the target day and the observed day. Lastly, we utilize the realization probability with the central limit theorem in order to set up the weight. These 3 WLS methods were implemented to improve the performance for both the GMLR and PMLR predictions.

Another way to improve our PMLR performance was to group the training set. Five grouping methods have been tested. Two of them are seasonal grouping methods simply using the date (G1 and G2) and three of them are non-seasonal grouping methods using the *k*-means clustering algorithm (G3, G4, and G5). The best prediction performance among them was from the G5 method, which divided the training set into as many clusters as possible with multiple selections. In addition, WLS-M3 method can be applied to the G5 grouping at the same time, which leads to the best prediction performance.

We have conducted two case studies with electric data from Austin, Texas (USA) and Jeju (South Korea) and compared the prediction performances of the GMLR and PMLR, solving the OLS or WLS problems to verify our arguments. It was preferable to apply the PMLR prediction rather than simply using the GMLR for a certain condition. Additionally, the performance could be further improved by the additional application of the proper WLS technique. We have seen similar results in the WLS experiments but somewhat different results in the grouping experiments during these two case studies. We have found that we should consider the climate characteristics to set the clustering index and decide other options in the grouping method. In conclusion, we have demonstrated that the performance could be improved by using the PMLR prediction instead of using the GMLR exclusively and further enhanced by the application of the proper WLS method and the optimal grouping method.

Bibliography

[CPIC96] T. Czernichow, A. Piras, K. Imhof, and P. Caire. Short term electrical load forecasting with artificial neural networks. *Engineering Intelligent Systems for Electrical Engineering and Communications*, 4(3):85 – 99, 1996.

[EMD06] Gonzalez-Romera E, Jaramillo-Moran M.A, and Carmona-Fernandez D. Monthly electric energy demand forecasting based on trend extraction. *IEEE Trans. Power Syst.*, 21(4):1946–1953, 2006.

[HHS04] H.M.Al-Hamadi and S.A. Soliman. Short-term electric load forecasting based on kalman filtering algorithm with moving window weather and load model. *Electric Power Systems Research*, 68(1):47–59, 2004.

[HPS01] H.S. Hippert, C.E. Pedreira, and R.C. Souza. Neural networks for short-term load forecasting: A review and evaluation. *IEEE Transactions on Power Systems*, 16(1):44 – 55, 2001.

[KARM98] A. Khotanzad, R. Afkhami-Rohani, and D. Maratukulam. Ann stlf artificial neural network short-term load forecaster generation three. *IEEE Transactions on Power Systems*, 13(4):1413 – 1422, 1998.

[MBB05] P. E. McSharry, S. Bouwman, and G. Bloemhof. Probabilistic forecasts of the magnitude and timing of peak electricity demand. *IEEE Trans. Power Syst.*, 20(2):1166–1172, 2005.

[MR89] I. Moghram and S. Rahman. Analysis and evaluation of five short-term load forecasting techniques. *IEEE Trans. Power Syst.*, 4(4):1484–1491, 1989.

[RH93] S. Rahman and O. Hazim. A generalized knowledge-based short-term load-forecasting technique. *IEEE Transactions on Power Systems*, 8(2):508 – 514, 1993.

[SIK14] Per Goncalves Da Silvar, Dejan Ilic, and Stamatis Karnouskos. The impact of smart grid prosumer grouping on forecasting accuracy and its benefits for local electricity market trading. *IEEE Transactions on Smart Grid*, 5(1):402–410, 2014.

[Whi80] Halbert White. A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity. *Econometrica*, 48(4):817–838, 1980.