Tsinghua-Berkeley Shenzhen Institute
Learning from Data
Fall 2019

Final Report

Xiaoting Wang, Yiqian Wu December 30, 2019

A Gaussian Process Regression Based Approach for Predicting Building Cooling and Heating Consumption

Abstract

Load prediction is important for power system design which guarantees the safety, stability and economic efficiency. Gaussian process regression model is employed to predict the electricity consumption of the campus building GUND based on historical weather information. Zero mean function and SEard covariance function are chosen to perform Gaussian process. The prediction results showed high accuracy rate with R^2 values of train set and test set at 0.9947 and 0.8398, respectively. The proposed Gaussian process regression model is found feasible for the energy consumption prediction with the given dataset.

1 Introduction

Load prediction of power system is an important part for the planning of power system. It can guarantee the safety, stability and economic efficiency of power system [1]. To do load prediction, we start from historical data of electricity demand. Combining with the data related to weather, economy, population and so on, the prediction of future power demand can be derived. Precise and accurate load prediction would be helpful to the controlling and operation of power grid dispatch. To be more specific, we can place and plan the usage of battery energy storage system on the user side according to predicted electricity demand. Battery energy storage system has the ability of allocating and reserving power. By reasonably placing and planning it on the user side, the goals of saving the electricity cost for users and saving the energy for the whole human beings can be achieved at the same time.

In this project, the input to our algorithm is the cooling and heating electricity consumption data of a certain building called GUND at Harvard college as well as several related factors data which may affect usage of electricity consumption. Those factors include month, day, hour, weekday, occupancy, cooling degree, heating degree of air-conditioners, humidity, solar-radiation, air pressure as well as wind-speed. For example, different hours of the day may contribute to different number of students in the building. To hold the specific temperature for the building, the cooling and heating electricity consumption of the building must be adjusted. We use Gaussian process regression to output a predicted electricity consumption of a certain period of future time, based on the historical data.

2 Related work

Load forecasting methods can be roughly divided into two types. The first type is the model-based forecasting method represented by Time-series method and Regression analysis method, the latter of which fits historical load data with certain parameters. The second is the artificial intelligence methods represented by Expert systems, Neural networks, and Fuzzy logic systems.

The historical data of power load is a time series of sampling and recording according to a certain time interval. It has strong randomness. Literature [2] introduces the Time-series model. It is an effective method for dealing with random sequences. By analyzing historical load data information and establishing mathematical models, it comes up with the rules of the changing of random variables, and then determines mathematical expressions to predict future loads. Literature [3] describes a method based on Grey system theory. It mostly works well in the case of insufficient data. The GM (1,1) gray forecasting model can better simulate the exponential growth curve and obtain good forecasting accuracy. Recently, more researchers tend to combine several methods together to rise the accuracy of load prediction. In [4] , the load is decomposed into low frequency components and high frequency components by wavelet decomposition, and the law of each frequency component of the load is found out. The similar days of the load to be predicted are selected by the fuzzy gray correlation clustering method. Different neural networks are used for different frequency band loads. However, neural network is a typical global approximation network. Weights of the network have an impact on each output, resulting in a slow learning rate. In addition, the network has randomness in determining the weights, and the determination of the number of network layers and the number of hidden layer nodes lacks theoretical guidance. Gaussian process regression as a regression method in supervised learning [5] has been widely used in various fields [6, 7, 8]. Gaussian process regression has the advantages of easy implementation, probabilistic significance of output, and good adaptability to nonlinear problems. It is a non-parametric Bayesian machine learning technique that provides a flexible prior distribution over functions, enjoys analytical tractability, and has a fully probabilistic work-flow that returns robust posterior variance estimates, which quantify uncertainty in a natural way [7].

3 Dataset and features

In this project, the data we have is an integration data including all usage and environment conditions of a building called GUND in Harvard University between 2016/2/1 to 2016/4/30. All the data samples used in the model are on an hourly basis and it includes 2154 points. Every point consists of the electricity consumption at that hour and the corresponding features data. Those features include the information of month, day, hour, weekday, occupancy rate, cooling degrees and heating degrees of air-conditioners, humidity ratio(kg/kg), solar-radiation rate(W/m^2), air pressure($mbar$) and wind speed(m/s). The original data we received are weather data of several years, as well as the occupancy and electricity data of several years. We just pick three months and integrate them together into one excel profile. Some missing points were canceled. A part of the excel profile is shown below. (First column is just for time recording, for the reading convenience, so the xlsread of this excel starts from the second column.)

Table 1: Data sample of GUND

Date/Time	Month	Day	Hour	Weekday	Occupancy	Cooling degree	Heating degree	Humidity ratio	Solar radiation	Pressure	Wind speed	Electricity
2016/2/1 1:00	2	1	1	1	1	0	7.87	0.01	1.00	1004	1.46	142.51
2016/2/1 2:00	2	1	2	1	1	0	7.59	0.01	1.00	1004	2.02	139.26
2016/2/1 3:00	2	1	3	1	1	0	7.50	0.01	1.00	1003	1.75	136.13

We assign the first two-month data (the data of February and March) as training data and the third-month data as testing data (the data of April). The training accuracy and testing accuracy are obtained accordingly. Because this load prediction is the first step for the planning and operation of battery energy storage system which is the topic we are doing now in 1C laboratory. The data we used is from our supervisor who obtained the data from his colleagues he once worked with.

In the preprocessing step, data normalization is employed to ensure that all features are weighted equally, i.e. avoid biases, and speed up the calculations in gradient descent convergency. The features are normalized by the max mean value to the range of [-1, 1]:

$$x' = \frac{x}{|\max(x)|},$$

where x is an original value, x' is the normalized value.

4 Methods

In this project, Gaussian process regression is applied to predict the building cooling and heating electricity consumption. The GPML Toolbox, written by Carl Edward Rasmussen and Hannes Nickisch, is utilized to complete the relevant algorithms [9].

According to the study by Rasmussen and Williams, a Gaussian process is specified by a mean function $\mu(\mathbf{x})$ and a covariance function $k(\mathbf{x}, \mathbf{x}')$ consisting of a set of hyperparameters $\boldsymbol{\theta}$ respectively, which can be learned during the training process [10]. Accordingly, the prediction can be conducted with the learned hyperparameters:

$$f(\mathbf{x}) \sim \mathcal{GP}(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')).$$

Firstly, the form of the mean function and covariance function is determined. The mean function $\mu(\mathbf{x})$ reflects the expected function value at input $\mathbf{x}$:

$$\mu(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})].$$

In this project, The mean function $\mu(\mathbf{x})$ is set to 0 in order to simplify the computations.

The covariance function $k(\mathbf{x}, \mathbf{x}')$ models the dependence between the function values at different input points $\mathbf{x}$ and $\mathbf{x}'$:

$$k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - \mu(\mathbf{x}))(f(\mathbf{x}') - \mu(\mathbf{x}'))].$$

The covariance function takes two hyperparameters: a characteristic length-scale *ell* and the standard deviation of the signal s_f. A widely used covariance function in Gaussian process is the squared exponential (SE). However, SE assumes that a single, global characteristic length scale *ell* can appropriately evaluate proximity in all input dimensions. Since there are several features for input $\mathbf{x}$, a squared exponential with automatic relevance determination (SEard) covariance function is chosen in this project, whose length-scale parameters ell_i differs for every input dimensions:

$$k_{SEard}(\mathbf{x}, \mathbf{x}') = s_f^2 \exp(-\frac{(\mathbf{x}-\mathbf{x})'\Theta^{-1}(\mathbf{x}-\mathbf{x})}{2}),$$

where Θ is the diagonal matrix with diagonal elements ell_i.

Therefore, the predictive distribution of $f(x)$ is normal with mean $\mu(\hat{\mathbf{x}})$ and variavce $\sigma^2(\hat{\mathbf{x}})$ for a noise-free new input $\hat{\mathbf{x}}$:

$$\boldsymbol{\mu}(\hat{\mathbf{x}}) = \mathbf{K}(\mathbf{x}, \hat{\mathbf{x}})'(\mathbf{K}(\mathbf{x}, \mathbf{x}) + \sigma_n^2 \mathbf{I})^{-1}\mathbf{y}$$
$$\boldsymbol{\Sigma}(\hat{\mathbf{x}}) = \mathbf{K}(\hat{\mathbf{x}}, \hat{\mathbf{x}}) - \mathbf{K}(\mathbf{x}, \hat{\mathbf{x}})'(\mathbf{K}(\mathbf{x}, \mathbf{x}) + \sigma_n^2 \mathbf{I})^{-1}\mathbf{K}(\mathbf{x}, \hat{\mathbf{x}}),$$

where $\mathbf{K}$ is the covariance functions, σ_n^2 is the variance of Gaussian noise in training target $\mathbf{y}$.

Based on Bayes Formula, hyperparameters are trained by maximizing the marginal likelihood of the training data, i.e. minimizing the negative log marginal likelihood.

$$P(\hat{\mathbf{y}}|\hat{\mathbf{x}}, \mathbf{x}, \mathbf{y}) = \int P(\hat{\mathbf{y}}|\hat{\mathbf{x}}, \boldsymbol{\theta}, \mathbf{x}, \mathbf{y}) P(\boldsymbol{\theta}|\mathbf{x}, \mathbf{y}) d\boldsymbol{\theta} = \mathcal{N}(\hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}}, \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}} + \sigma_n^2 \mathbf{I})$$
$$\mathcal{L} = -\log P(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) = \frac{1}{2}\mathbf{y}'\boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1}\mathbf{y} + \frac{1}{2}|\boldsymbol{\Sigma}_{\boldsymbol{\theta}}| + \frac{n}{2}\log 2\pi$$
$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} = \frac{1}{2} tr(\boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1} \frac{\partial \boldsymbol{\Sigma}_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}}) - \frac{1}{2}\mathbf{y}'\boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1} \frac{\partial \boldsymbol{\Sigma}_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}} \boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1}\mathbf{y}$$

Relevant algorithm is shown as follows:

```
%% Train
% The covfunc, likfunc and gp are from GPML Toolbox
% covSEard: k(x^p,x^q) = sf^2 * exp(-(x^p - x^q)'*inv(P)*(x^p - x^q)/2)
covfunc = @covSEard;
% likGauss: likGauss(t) = exp(-(t-y)^2/2*sn^2) / sqrt(2*pi*sn^2)
likfunc = @likGauss;

% Initialize hyperparameters
% The covSEard takes n+1 parameters: log of ell_i and log of sf
% The mean function is empty
hyp.cov = log(1) * ones((size(train.X, 2) + 1),1);
hyp.lik = log(100);
hyp.mean = [];

% Train hyperparameters
% The minimize function returns the solution
% parameters
%   hyp         initial guess
%   @gp         the function to be minimized. The function return
%               two arguments: the value of the function and the
%               partial derivatives wrt hyp
%   -3000       maximum length of the run
%   others      parameters passed to gp
%   @infExact   the inference method
```

```
hyp = minimize(hyp, @gp, -3000, @infExact, [], covfunc, likfunc, train.X,
    train.Y);

% Output predictive mean to test the train accuracy
% parameters
%   train.m     predictive output means
%   train.s2    predictive output variances
%   train.X     training inputs
%   train.Y     training targets
%   train.X     test inputs
[train.m, train.s2] = gp(hyp, @infExact, [], covfunc, likfunc, train.X,
   train.Y, train.X);
```

Secondly, prediction is conducted with the trained hyperparameters. Relevant algorithm is shown as follows:

```
%% Predict
% parameters
%   test.m     predictive output means
%   test.s2    predictive output variances
%   train.X    training inputs
%   train.Y    training targets
%   test.X     test inputs
[test.m, test.s2] = gp(hyp, @infExact, [], covfunc, likfunc, train.X,
   train.Y, test.X);

% Reverse normalization
train.m = train.m * scale.Y;
test.m = test.m * scale.Y;
test.s2 = test.s2 * scale.Y * scale.Y;
train.Y= train.Y * scale.Y;
test.Y = test.Y * scale.Y;
```

5 Results and discussion

In this project, the Gaussian process prediction of building cooling and heating consumption is conducted based on the historical weather information in February and March. The prediction results made by Gaussian process is stated in the form of probabilistic distribution. Therefore, the comparison with the train set is provided by predictive mean, which is presented in Figure 1.

In order to evaluate the accuracy of the prediction, the coefficient of determination R^2 is measured. In regression, R^2 demonstrates the extent of how well the predictions approximate the observed data. Normally, R^2 ranges from 0 to 1. An R^2 of 1 indicates a perfect fit to the data. R^2 can be calculated as follows:

$$R^2 = 1 - \frac{\Sigma(y_i - \hat{y}_i)^2}{\Sigma(y_i - \bar{y})^2}.$$

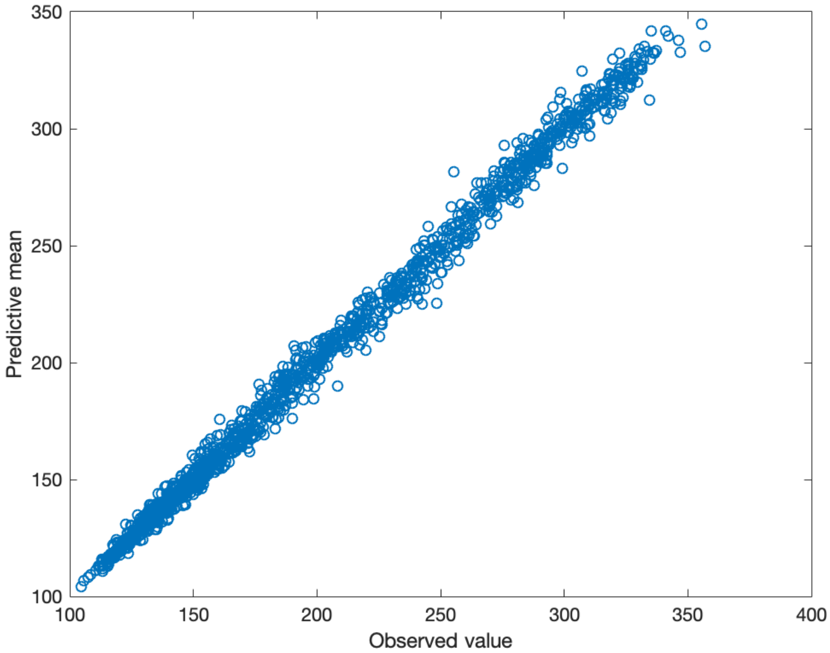

Figure 1: Prediction results for train set

where y is the observed value, $\hat{y}$ is the predicted value, $\bar{y}$ is the mean value of the observed value.

According to the prediction results, the R^2 value of the train set reaches 0.9947. In this sense, it can be concluded that the utilized Gaussian process model shows good predictive ability on the training data.

Apply the Gaussian process model to the test data, and the results is depicted in Figure 2. The red solid line indicates the predictive mean, the blue area presents a 95% confidence region of the prediction results, the observed values are described as dots. As shown in Figure 2, the majority of the observed values are in the 95% confidence region. Additionally, the R^2 value of the test set reaches 0.8398.

As we discussed in the method part, the result of Gaussian process regression can be shown as a probability form, which is the biggest difference from the other supervised method. This means that the target values of input testing data are not some fixed value. Instead, by changing the multiplication parameters of mean and variance of target y, different confidence region can be obtained.

In the planning strategy of battery energy storage system problem, we could use the load prediction region from Gaussian process regression to come up with strategies with different conservative levels. In other words, if we want to do a more conservative planning of battery energy storage system, we would utilize the higher bound of the predictive region as the predictive load value. This could guarantee that the capacity of battery energy storage system could be enough for users even if the actual load of users is really high in real-time operation. This is also one of the advantages of Gaussian process regression for load prediction.

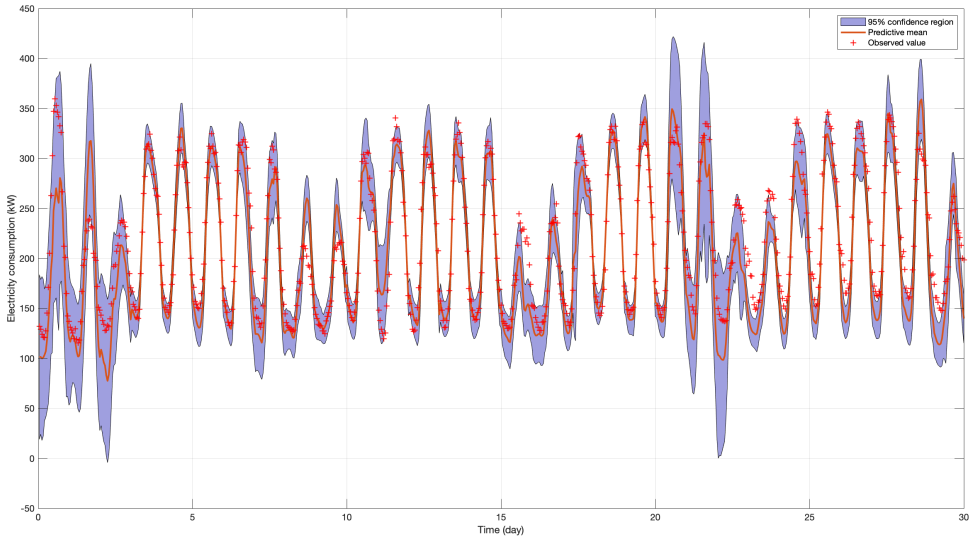

Figure 2: 1-month prediction of electricity consumption

6 Conclusion

In this project, we successfully showed that the kernel-based Gaussian process regression model we proposed can predict electricity consumption of a certain building with high accuracy rate. Firstly, several relevant features are selected to predict electricity consumption of the campus building GUND. In the preprocessing step, data normalization is conducted to avoid biases and speed up the calculations in gradient descent convergency. Secondly, zero mean function and SEard covariance function are chosen to perform Gaussian process. The corresponding hyperparameters of the utilized covariance function are trained by maximizing the marginal likelihood function. Finally, prediction is conducted with the trained hyperparameters.

The prediction results show that the majority of the predictive mean are in the 95% confidence region of the observed values. The accuracy of prediction is also evaluated by R^2 value. The R^2 values of train set and test set are 0.9947 and 0.8398, respectively. Therefore, the proposed Gaussian process regression model is found feasible for the energy consumption prediction, and additional comparison to other machine learning methods will be conducted to demonstrate the effectiveness of Gaussian process in the future.

A Source code

Source code for this project is shown as follows:

```
clear all;
clc;

%% Load data
rawtemp = csvread('DataforGP_Gund.csv', 1, 1);

temp=rawtemp(rawtemp(:,12)>0,:);

month =temp(:,1);
day = temp(:,2);
hour = temp(:,3);
weekday = temp(:,4);
occupancy = temp(:,5);
CoolD = temp(:,6);
HeatD = temp(:,7);
HumiRatio=temp(:,8);
Solar=temp(:,9);
Pressure=temp(:,10);
Wind=temp(:,11);

totalElectricity = temp(:,12);

Y = totalElectricity;

X = [day, hour, weekday, occupancy, CoolD, HeatD, HumiRatio, Solar,
    Pressure, Wind];

%% Preprocess data
% Split train and test sets

train.index = (month <= 3);
test.index = (month > 3);

train.X = X(train.index, :);
train.Y = Y(train.index);

test.X = X(test.index, :);
test.Y = Y(test.index);

% Normalize data to [-1, 1]
scale.Y = max(abs(train.Y));
scale.X = max(abs(train.X));

numTrain = length(train.Y);
numTest = length(test.Y);
```

```
test.X = test.X ./ scale.X;
test.Y = test.Y ./ scale.Y;
train.X = train.X ./ scale.X;
train.Y = train.Y ./ scale.Y;

%% Train
covfunc = @covSEard;
likfunc = @likGauss;

% Initialize hyp
hyp.cov = log(1) * ones((size(train.X, 2) + 1),1);
hyp.lik = log(100);
hyp.mean = [];

% Train hyperparameters
hyp = minimize(hyp, @gp, -3000, @infExact, [], covfunc, likfunc, train.X,
    train.Y);

% Output predictive mean
[train.m, train.s2] = gp(hyp, @infExact, [], covfunc, likfunc, train.X,
   train.Y, train.X);

% Output training accuracy (R2)
train_accuracy = detcoeff(train.Y, train.m);

%% Predict
[test.m, test.s2] = gp(hyp, @infExact, [], covfunc, likfunc, train.X,
   train.Y, test.X);

% Reverse normalization
train.m = train.m * scale.Y;
test.m = test.m * scale.Y;
test.s2 = test.s2 * scale.Y * scale.Y;
train.Y= train.Y * scale.Y;
test.Y = test.Y * scale.Y;

% plot
figure(1)
plot(train.Y, train.m, 'o');
xlabel('Observed value'),ylabel('Predictive mean');

figure(2)
f = [test.m + 1.96*sqrt(test.s2); flipdim(test.m - 1.96*sqrt(test.s2), 1)
   ]; % Confidence region: 95%=1.96, 98%=2.326, 99%=2.576
test_m_up=test.m+1.96*sqrt(test.s2);test_m_low=test.m-1.96*sqrt(test.s2);
z = [1 : numTest]';
fill([z; flipdim(z,1)], f, [5 5 7]/8);
```

```
hold on;
plot(test.m, '-','LineWidth', 2);
plot(test.Y, '+r', 'MarkerSize', 6);
grid on;
hold off;
legend('95% confidence region','Predictive mean','Observed value');
xlabel('Time (day)'),ylabel('Electricity consumption (kW)');
set(gca,'XTick',0:24*5:30*24);
set(gca,'XTickLabel',{'0','5','10','15','20','25','30'})
xlim([0 30*24])

%% Accuracy (R2)
train_accuracy
test_accuracy = detcoeff(test.Y, test.m)
```

References

[1] Yuan Zhang and Mo-Yuen Chow. Microgrid cooperative distributed energy scheduling (codes) considering battery degradation cost. In *2016 IEEE 25th International Symposium on Industrial Electronics (ISIE)*, pages 720–725, 2016.

[2] Ibrahim Moghram and Saifur Rahman. Analysis and evaluation of five short-term load forecasting techniques. *Power Engineering Review, IEEE*, 9:42–43, 12 1989.

[3] Wei Li, Weidong Dong, and Yanan Yuan. Load forecasting based on discrete grey model optimized by composite function and genetic algorithm. *Electric Power Automation Equipment*, 32:76–79, 04 2012.

[4] Xueping Pan, Ping Zhang, and Wenchao Xue. Short-term load forecasting based on fuzzy clustering wavelet decomposition and bp neural network. *Dianli Zidonghua Shebei/Electric Power Automation Equipment*, 32:1–4, 03 2011.

[5] Marc Peter Deisenroth, Dieter Fox, and Carl Edward Rasmussen. Gaussian processes for data-efficient learning in robotics and control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37:408–423, 02 2015.

[6] Yu Lei, Danning Zhao, Yuping Gao, and Hongbing Cai. Prediction of length-of-day variations based on gaussian processes. *Chinese Astronomy and Astrophysics*, 39:368–379, 07 2015.

[7] Maziar Raissi, Hessam Babaee, and George Karniadakis. Parametric gaussian process regression for big data. *Computational Mechanics*, 64, 05 2019.

[8] Lukas Hewing and Melanie Zeilinger. Cautious model predictive control using gaussian process regression. *IEEE Transactions on Control Systems Technology*, PP, 05 2017.

[9] Carl Edward Rasmussen and Hannes Nickisch. Gaussian processes for machine learning (gpml) toolbox. *Journal of Machine Learning Research*, 11(11):3011, November 2010.

[10] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Process for Machine Learning*. The MIT Press, 2006.