Data Estimation and Inference Lab – Gaussian Process Regression

Chia-Man Hung
CDT AIMS, University of Oxford
chia-man.hung@eng.ox.ac.uk

Abstract

Gaussian process is a classical non-parametric statistical model suitable for solving regression problems. In this lab, we experiment with different kernels to fit and predict time series data derived from a weather sensor network. This short report aims at summarizing our approaches, focusing on the actual implementation and experiments.

1. Introduction

In this lab, given weather sensor data from the Port of Southampton, we are interested in predicting missing measurements of quantities such as tide height and air temperature. More details on the data can be found in the lab sheet[1]. The performance is evaluated by the root mean square error between the prediction and the ground truth. Since we are only given the ground truth of tide height, our experiments are only conducted on tide height. The goal of the lab is to learn how to implement Gaussian Process from scratch. For this reason, external machine learning libraries are not used. In the following, we first describe Gaussian Process and the algorithm to fit the model and predict missing values. Then, we explain the experiments that are conducted.

2. Gaussian Process

2.1. Algorithm

A very comprehensive introduction to Gaussian Process Regression is provided in [2]. In practice, we imagine our data set as a single point sampled from some multivariate Gaussian distribution. We define a covariance kernel function k to model the behavior between two data points. Without prior knowledge, it is common to assume the Gaussian distribution to have 0 as its mean. Noise is sometimes included in the kernel function. Assume we have n observations of $\{(t_i, y_i)\}_{i=1\ldots n}$ and we want to predict the missing value on t_*. We first compute the three following matrices.

$$K = \begin{bmatrix} k(t_1,t_1) & k(t_1,t_2) & \cdots & k(t_1,t_n) \\ k(t_2,t_1) & k(t_2,t_2) & \cdots & k(t_2,t_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(t_n,t_1) & k(t_n,t_2) & \cdots & k(t_n,t_n) \end{bmatrix} \tag{1}$$

$$K_* = \begin{bmatrix} k(t_*,t_1) & k(t_*,t_2) & \cdots & k(t_*,t_n) \end{bmatrix} \tag{2}$$

$$K_{**} = k(t_*,t_*) \tag{3}$$

[1] http://www.robots.ox.ac.uk/~mosb/teaching/AIMS_CDT/CDT_estimation_inference_lab.pdf

Our assumption of Gaussian distribution leads to

$$\begin{bmatrix} \mathbf{y} \\ y_* \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, \begin{bmatrix} K & K_*^T \\ K_* & K_{**} \end{bmatrix}). \tag{4}$$

The conditional probability $y_*|\mathbf{y}$ follows a Gaussian distribution.

$$y_*|\mathbf{y} \sim \mathcal{N}(K_*K^{-1}\mathbf{y}, K_{**} - K_*K^{-1}K_*^T) \tag{5}$$

The best estimate for y_* is thus

$$\bar{y_*} = K_*K^{-1}\mathbf{y}. \tag{6}$$

The uncertainty is captured by

$$\mathrm{var}(y_*) = K_{**} - K_*K^{-1}K_*^T. \tag{7}$$

We follow Algorithm 2.1 in [3] to compute predictive means and variance and log marginal likelihood. The implementation addresses the covariance matrix inversion using Cholesky decomposition. Once the decomposition done, solving triangular systems is relatively simple.

2.2. Kernels

Below is a list of kernels we tried in modeling the similarity between two data points to compute the covariance matrix in Gaussian Process. Chapter 2 Expressing Structure with Kernels[2] of [1] provides a good reference for the choice of kernels.

- **Exponentiated quadratic** $k(t_1,t_2) = \exp(\frac{-(t_1-t_2)^2}{2l^2})$
- **Rational quadratic** $k(t_1,t_2) = (1 + \frac{1}{2\alpha}(\frac{t_1-t_2}{l})^2)^{-\alpha}$
- **Periodic** $k(t_1,t_2) = \exp(-\frac{2(\sin(\pi(t_1-t_2)/\rho))^2}{l})$
- **Matern 3/2** $k(t_1,t_2) = (1 + \sqrt{3}\frac{|t_1-t_2|}{l})\exp(-\sqrt{3}\frac{|t_1-t_2|}{l})$

Noise is added to the kernel function as $(t_1,t_2) \mapsto \sigma_n^2\delta(t_1,t_2)$. This also ensures that the covariance matrix K is positive definite and thus invertible.

2.3. Hyperparameters Selection

To choose the hyperparameters l, α, ρ, σ introduced in kernel functions, we maximize the log likelihood.

2.4. Global vs Sequential Prediction

In global prediction, we fit our model with all the observations and use it to predict all missing values. In sequential prediction, we fix a lookahead and predict each missing value at time t by using the model fit with the observations in time $[t - \text{lookahead}, t)$.

[2] The kernel cookbook available at http://www.cs.toronto.edu/~duvenaud/cookbook/

3. Experiments and Results

Our implementation [3] is done in matlab. The main code to execute is named *GPmain.m*. Some parts of the code are factored out to functions for readability – *kernels.m*, *calc_mean_var_ll.m*. *fill_mean_err.m* and *plot_prediction.m* are used for plotting and visualization.

3.1. Data observation

In the Sotonmet[4] data, there are 1259 rows, in which the first one is the header. We observe that the update duration is always 0; the update date and time and the reading date and time is always the same. Thus, we only keep the reading date and time. Line 551 is removed since it has the same reading date and time as the previous line, but different values for the other measurements, which is regarded as a data error. As a quick note, among the 1257 remaining rows, 341 rows have missing values. For rows in which the tide height is missing, the five measurements air pressure, air temperature, wind direction, wind gust speed, wind speed are also missing, i.e. the only measurement that helps our prediction is the time. The following experiments only use columns 3, 6, 11, which are reading date and time, tide height, true tide height respectively. It becomes a time series regression problem. The true tide height is only used for evaluation purpose.

3.2. Experiments

First, we experimented global prediction with different kernel functions. Results are shown in Figures 1, 2, 3, 4. When the noise is weighted more heavily, the more uncertain we are about our prediction. We keep weight of the noise constant for different kernel functions. Maximizing the log likelihood sometimes over-fits the data. Periodic kernel seems to give the best result, since our data is more or less periodic. Matern 3/2 kernel and rational quadratic kernel are comparable to periodic kernel. Prediction by Matern 3/2 kernel has large variance in place where there are fewer observations.

Second, we experimented sequential prediction with different lookahead. Results are shown in Figures 5, 6, 7, 8, 9. We keep the hyperparameters constant as it is computationally expensive to solve an optimization problem at each value to predict. As expected, fewer the observations in the previous time interval are, larger the variance is. Similarly, the variance gradually increases when the observations become sparse. The larger the lookahead is, the less impact it has when incrementing it.

4. Conclusion and Future Work

In this report, we have summarized the algorithm used to perform Gaussian Process Regression. We have implemented it to perform prediction in a real-world problem. Different kernels in global prediction have been compared and the impact of different lookahead in sequential prediction has been discussed. For further work, we could also explore using different mean functions in Gaussian distribution or combining different kernel functions.

References

[1] D. Duvenaud. *Automatic model construction with Gaussian processes*. PhD thesis, University of Cambridge, 2014.

[3] The source code can be found at `https://github.com/ascane/gaussian-process`.

[4] `http://www.robots.ox.ac.uk/~mosb/teaching/AIMS_CDT/sotonmet.txt`

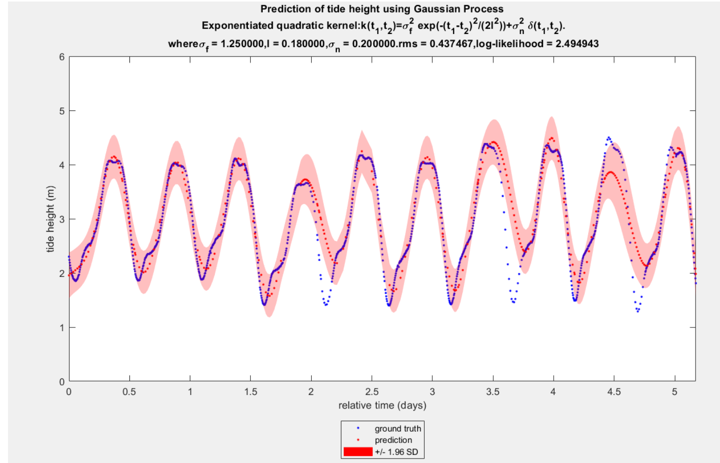

Figure 1. Global prediction with Exponentiated quadratic kernel. $k(t_1, t_2) = \sigma_f^2 \exp(\frac{-(t_1-t_2)^2}{2l^2}) + \sigma_n^2 \delta(t_1, t_2)$, where $\sigma_f = 1.25, l = 0.18, \sigma_n = 0.2$. log likelihood $= 2.49$. root mean square error $= 0.43$.

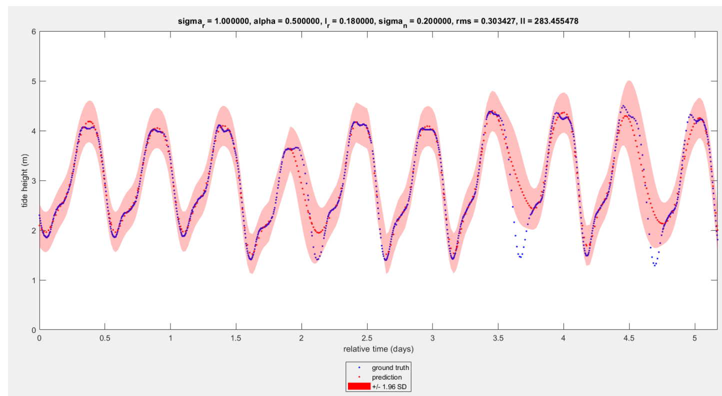

Figure 2. Global prediction with Rational quadratic kernel. $k(t_1, t_2) = \sigma_r^2 (1 + \frac{1}{2\alpha}(\frac{t_1-t_2}{l})^2)^{-\alpha} + \sigma_n^2 \delta(t_1, t_2)$, where $\sigma_r = 1.0, \alpha = 0.5, l = 0.18, \sigma_n = 0.2$. log likelihood $= 283.46$. root mean square error $= 0.30$.

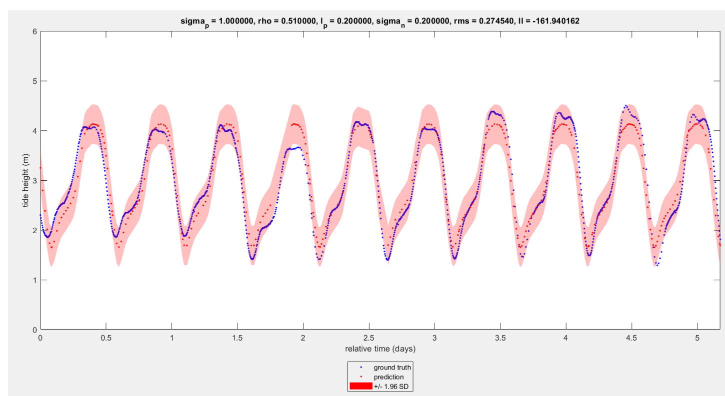

Figure 3. Global prediction with Periodic kernel. $k(t_1, t_2) = \sigma_p^2 \exp(-\frac{2(\sin(\pi(t_1-t_2)/\rho))^2}{l}) + \sigma_n^2 \delta(t_1, t_2)$, where $\sigma_p = 1.0, \rho = 0.51, l = 0.2, \sigma_n = 0.2$. log likelihood $= -161.94$. root mean square error $= 0.27$.

[2] M. Ebden et al. Gaussian processes for regression: A quick introduction. *The Website of Robotics Research Group in Department on Engineering Science, University of Oxford*, 2008.

[3] C. E. Rasmussen and C. K. Williams. *Gaussian processes for machine learning*, volume 1. MIT press Cambridge, 2006.

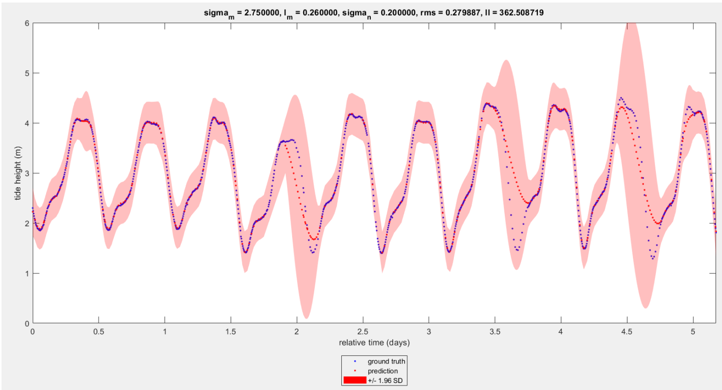

Figure 4. Global prediction with Matern 3/2 kernel. $k(t_1, t_2) = \sigma_m^2(1 + \sqrt{3}\frac{|t_1-t_2|}{l})\exp(-\sqrt{3}\frac{|t_1-t_2|}{l}) + \sigma_n^2\delta(t_1, t_2)$, where $\sigma_m = 2.75, l = 0.26, \sigma_n = 0.2$. log likelihood $= 362.51$. root mean square error $= 0.28$.

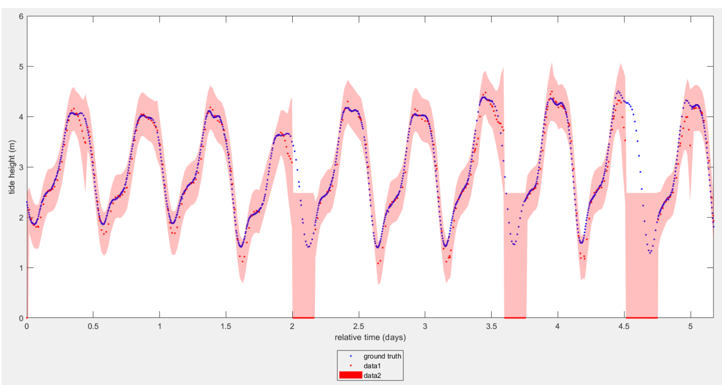

Figure 5. Sequential prediction with Exponentiated quadratic kernel. $k(t_1, t_2) = \sigma_f^2 \exp(\frac{-(t_1-t_2)^2}{2l^2}) + \sigma_n^2\delta(t_1, t_2)$, where $\sigma_f = 1.25, l = 0.18, \sigma_n = 0.2$. lookahead = 0.1. root mean square error $= 1.0499$.

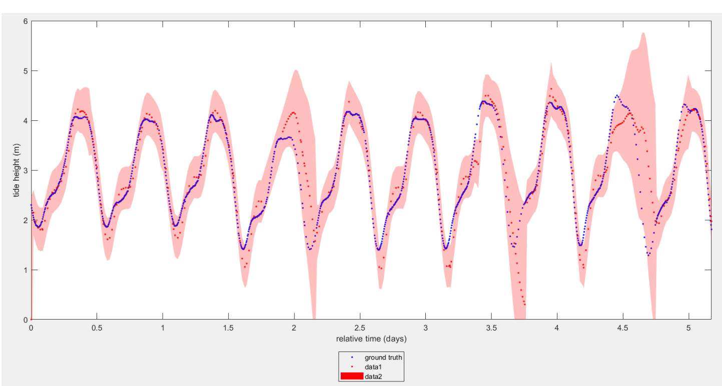

Figure 6. Sequential prediction with Exponentiated quadratic kernel. $k(t_1, t_2) = \sigma_f^2 \exp(\frac{-(t_1-t_2)^2}{2l^2}) + \sigma_n^2\delta(t_1, t_2)$, where $\sigma_f = 1.25, l = 0.18, \sigma_n = 0.2$. lookahead = 0.5. root mean square error $= 0.5055$.

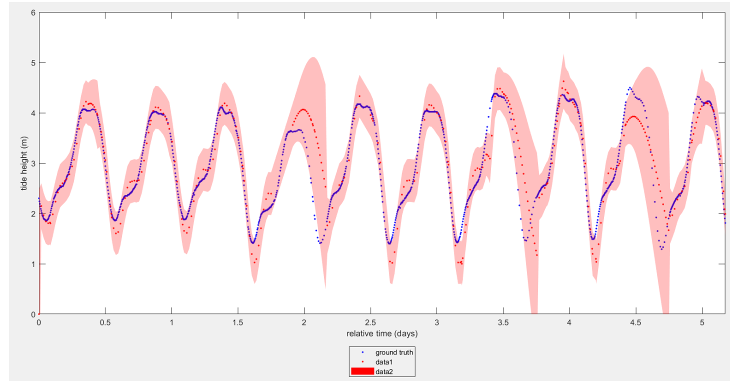

Figure 7. Sequential prediction with Exponentiated quadratic kernel. $k(t_1, t_2) = \sigma_f^2 \exp(\frac{-(t_1-t_2)^2}{2l^2}) + \sigma_n^2\delta(t_1, t_2)$, where $\sigma_f = 1.25, l = 0.18, \sigma_n = 0.2$. lookahead = 1. root mean square error $= 0.4951$.

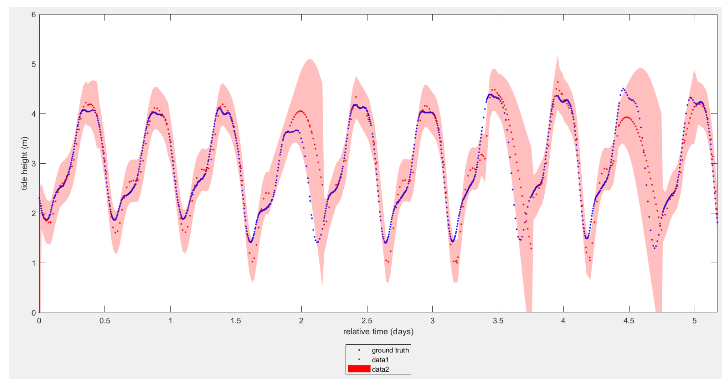

Figure 8. Sequential prediction with Exponentiated quadratic kernel. $k(t_1, t_2) = \sigma_f^2 \exp(\frac{-(t_1-t_2)^2}{2l^2}) + \sigma_n^2\delta(t_1, t_2)$, where $\sigma_f = 1.25, l = 0.18, \sigma_n = 0.2$. lookahead = 2. root mean square error $= 0.4950$.

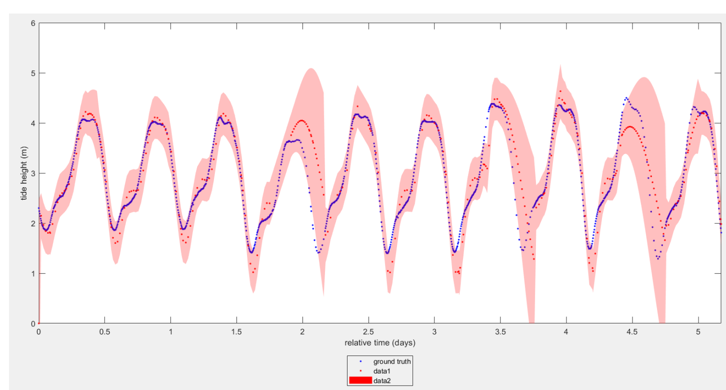

Figure 9. Sequential prediction with Exponentiated quadratic kernel. $k(t_1, t_2) = \sigma_f^2 \exp(\frac{-(t_1-t_2)^2}{2l^2}) + \sigma_n^2\delta(t_1, t_2)$, where $\sigma_f = 1.25, l = 0.18, \sigma_n = 0.2$. lookahead = 6. root mean square error $= 0.4950$.