CS 189: Introduction to Machine Learning **Spring 2014**

Homework 1 — Due Feb 25

Student: Ankush Gupta

Introduction

In this lab we investigate using Gaussian Processes for regression on weather data prediction. Following are the main features of this implementation.

- The code was implemented in Python.
- Two mean functions were analyzed : (1) constant mean = mean of the data (2) cubic spline fit to the data.[1]
- Two covariance functions were analyzed :
  1. Squared-Exponential = $\sigma^2 exp\left(-\frac{(x_1 - x_2)^T M^{-1}(x_1 - x_2}{2}\right) + \sigma_y^2 I$
  2. Periodic = $\sigma^2 exp\left(-l^2 \sin^2(2\pi/b|x_1 - x_2|)\right)$
- The hyperparameters were tuned by maximizing the marginal likelihood. Analytical gradients were used with the conjugate gradient optimization method.
- Sequential predictions were also carried out for Tide Height and Temperature time series.

Please note all figures indicate intervals of 1 and 2 standard deviations.

RMS Error with Different Means/ Covariances

The table below indicates the root-mean-squared error of the **tide-height** predcitions with respect to the ground-truth data. The covariance hyper-parameters used for these experiments were optimised through marginal likelihood as detailed in the next section.

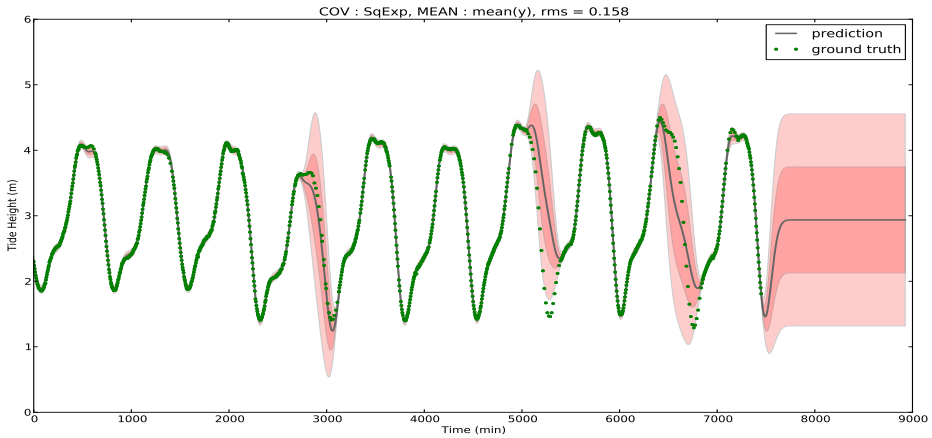
Mean \Covariance	Squared-Exponent	Periodic
Mean(data)	**0.158**	0.175
Spline	0.240	0.290

Table 1.1: RMS Error for tide-height predictions.

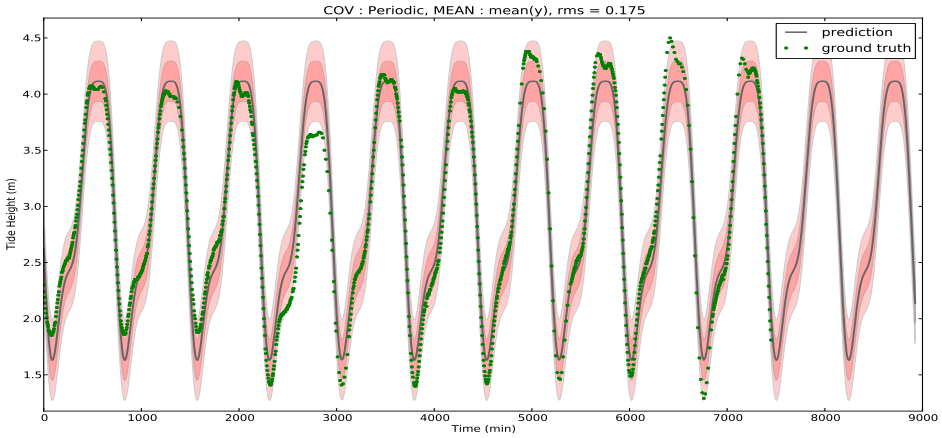
Note that the root-mean squared error is the lowest for squared-exponential with the mean function equal to the mean of the output. However, if we observe the plots in Figure 1.1, we note that the variance in the predictions in the region with no observations (beyond t=7000), is lower for the periodic covariance. This is because the periodic covariance is able to capture the long-term

[1] Note that these are strictly not bayesian methods as we are using the data itself to specify the prior.

periodic trend in tide-height data, which the squared-exponent is unable to; the square-exponent predicitions converge to the estimate of the mean function with the variance being equal to the sum of signal variance and observation variance.



(a)



(b)

Figure 1.1: The figure (a) above uses the square-exponential covariance where as figure (b) uses periodic (sin-squared-exponent). Even though the rms error of (a) is lower than (b) (0.158 ¡ 0.175), notice the variance towards the end (after $t > 7000$) — the variance for the squared-exponent is higher than periodic kernel because it cannot capture the long-term periodic trend in the data.

Below is a table similar to Table 1.1, for the **temperature data series**. Since there is no long-term periodic trend, only the square-exponent kernel was used. We note that the constant mean function achieves a lower rms error.

Determining the Hyper-parameters

The hyper parameters were determined by minimising the negative marginal log likelihood of the data. The gradient of negative-log-likelihood (NLL) was calculated analytically using the expression $= \frac{1}{2}Tr\left((K_y^{-1}yy^TK_y^{-T} - K_y^{-1})\frac{\partial K_y}{\partial \theta_j}\right)$. The actual optimisation was done using the the Conjugate-Gradient method of the`scipy.optimize.minimize` function.

Figure 1.2 shows the NLL of the tide-height data as a function of $\log(b)$ and $\log(\sigma_n)$ — the hyper-parameters in the square-exponent kernel $K(x_1, x_2) = \sigma^2 exp(-1/2|(x_1 - x_2)/b|) + \sigma_n^2 I$. In this plot, σ is held constant at its optimal value (which was found by minimising jointly over all three hyper-parameters). Gradient descent correctly found the minimum indicated in the figure.

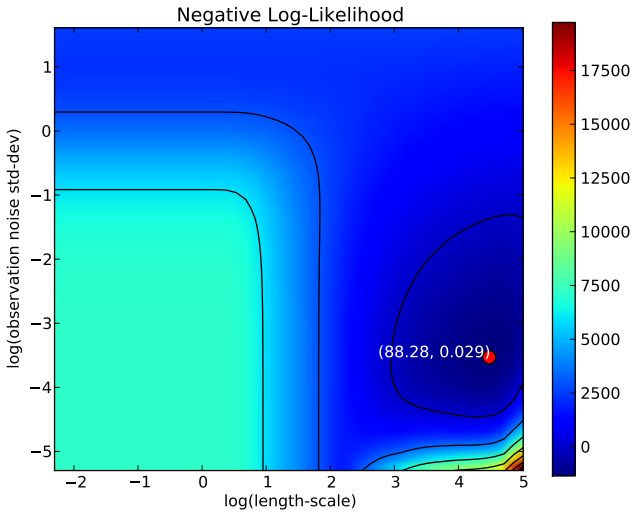


Figure 1.2: Negative-log-likelihood of tide-height data-series as a function of two of the hyper parameters of the square-exponential kernel. Red point marked achieves the minimum.

Mean \Covariance	Squared-Exponent
Mean(data)	**0.541**
Spline	1.954

Table 1.2: RMS Error for temperature predictions.

Sequential Prediction

Figure 1.3 shows the result of making predictions sequentially. The key-feature here is that the variance incases as the time-gap since the last observations increases and suddenly decreases when an observations is made.

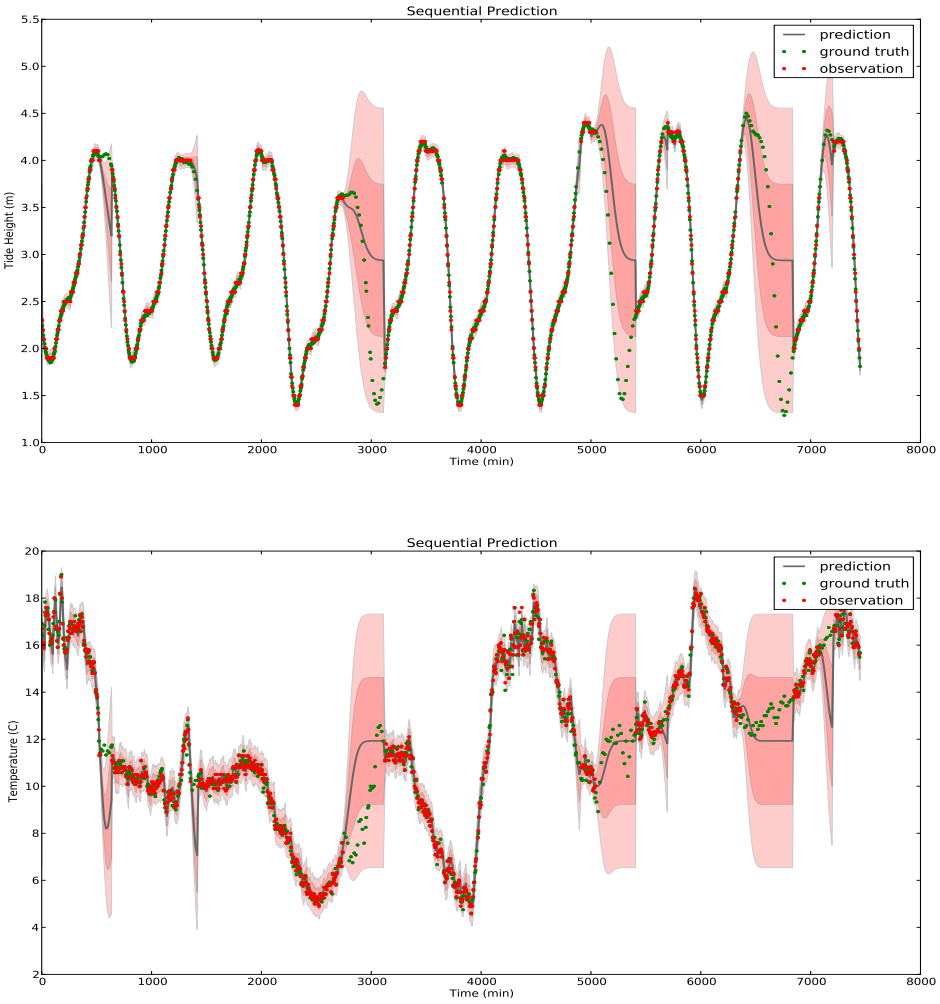


Figure 1.3: **Sequential prediction** : [Top] Tide Height. [Bottom] Temperature.

1.0.1 Future Consideration

I tried to get a generic class working — one which could taking any two existing kernels and produce a new one by multiplying them together. This could be used in capturing the local periodicity in the temperature data by multiplying the squared-exponential and the periodic kernel. Unfortunately, I faced numerical problems when optimising for hyper-parameters — which could not be resolved in the time I had. I would like to work on this further in the future and see if it can improve the performance.