Homework #3

Yongjin Shin
20090488
Industrial Management Engineering, POSTECH
dreimer@postech.ac.kr

September 3, 2018

Problem 1. Show detailed derivation of GP regression algorithm.

Solution The definition of a Gaussian process on a set S is a set of random variables $Z = (Z_x)_{x \in S}$ on a common probability space such that $\forall n \in \mathbb{N}, \forall x_1, ..., x_n \in S$, the vector $(Z_{x_1}, Z_{x_2}, ..., Z_{x_n})$ has a normal distribution. (In our textbook, $f(x)$ is used instead of Z_x.)

Let S be any index set and let $M : S \to \mathbb{R}$ be any function. Let $K : S \times S \to \mathbb{R}$ be a positive semi definite kernel, i.e. K is symmetric and $(K(x_i, x_j))_{i,j \leq n}$ is a p.s.d matrix $\forall n \geq 1$ and every $x_1, x_2, ..., x_n \in S$. Then by the theorem 2 in [1] there exists a probability space $(\Omega, \mathbb{F}, \mathbf{P})$ and random variables $Z_x : \Omega \to \mathbb{R}$ for each $x \in S$ such that $Z = (Z_x)_{x \in S}$ is a centered Gaussian process with mean function M, i.e. $\mu(x) := \mathbb{E}(Z_x)_{x \in S}$ and covariance function K, i.e. $k(s, t) := cov(Z_s, Z_t)_{\forall s,t \in S}$.

Now we know that with there exists a Gaussian process when it has any mean function and covariance function which has a p.s.d matrix. Then consider a training set $\mathbf{D} = \{(x_i, y_i), i = 1, ..., N\}$, with two cases; noise-free observations and noise in observed output values. Given a test set X_* of size $N_* \times D$, we want to predict the outputs Z_*.

Noise-free observations

First we will find the joint distribution of $[Z, Z_{x_*}]^T$, and then use the conditioning rule for Gaussian to compute the conditional distribution of $Z_{x_*}|Z$. By definition of the GP, the joint distribution of $[Z, Z_{x_*}]^T$ has following form:

$$\begin{pmatrix} \mathrm{Z} \\ \mathrm{Z}_* \end{pmatrix} \sim \begin{pmatrix} \mu \\ \mu_* \end{pmatrix} \begin{pmatrix} \mathrm{K} & \mathrm{K}_* \\ \mathrm{K}_*^T & \mathrm{K}_{**} \end{pmatrix}$$

where $K = k(X, X)$ is $N \times N$, $K_* = k(X, X_*)$ is $N \times N_*$, and $K_{**} = k(X_*, X_*)$ is $N_* \times N_*$. By the standard rules for conditioning Gaussian in [2] of chapter 4.3 eq(4.69), the posterior has the following form:

$$\begin{aligned} p(Z_{x_*}|x_*, X, Z) &= \mathcal{N}(Z_{x_*}|\mu_*, \Sigma_*) \\ \mu_* &= \mu(x_*) + K_*^T K^{-1}(Z - \mu(X)) \\ \Sigma_* &= K_{**} - K_*^T K^{-1} K_* \end{aligned}$$

Noise in observed output values

In this case, we should consider that $y = Z_x + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma_y^2 I)$. Since underlying random variables Z_x and error terms ϵ are independent, the distribution of a sum of two normally distributed forms another normal distribution with simple addition of mean and variance respectively[3]. Therefore $y \sim \mathcal{N}(0, K + \sigma_y^2 I)$.

From now on, we are going to assume zero-mean for the sake of notational simplicity. Hence, by definition of the GP, the joint distribution of the observed data and noise-free values on the test points is given as the following form:

$$\begin{pmatrix} \mathrm{y} \\ \mathrm{Z}_* \end{pmatrix} \sim \begin{pmatrix} 0 \\ 0 \end{pmatrix} \begin{pmatrix} \mathrm{K}_y & \mathrm{K}_* \\ \mathrm{K}_*^T & \mathrm{K}_{**} \end{pmatrix}$$

where $K + \sigma_y^2 I$ is denoted as K_y. The predictive density is

$$\begin{aligned} p(Z_{x_*}|x_*, X, y) &= \mathcal{N}(Z_{x_*}|\mu_*, \Sigma_*) \\ \mu_* &= K_*^T K_y^{-1}(y - \mu(y)) = K_*^T K_y^{-1} y \\ \Sigma_* &= K_{**} - K_*^T K_y^{-1} K_* \end{aligned}$$

In Gaussian Process, the covariance matrix might also be called the Gramm matrix, a smoother matrix, or simply the covariance. It is of the form:

$$\mathcal{C} = \begin{pmatrix} \mathrm{k}_{ii} & \dots & \mathrm{k}_{iN} \\ \vdots & \ddots & \vdots \\ \mathrm{k}_{Ni} & \dots & \mathrm{k}_{NN} \end{pmatrix}$$

A Gramm matrix is symmetric and positive semi definite if and only if there exists a feature map $\varphi(x)$ such that $k(x, x') = \varphi(x)\varphi(x')$. Where $\varphi(x)$ can be infinite dimensional.This means that a Gramm matrix may be equivalently represented by either a kernel function or by a basis function. As a commonly used kernel function, called *squared exponential kernel*, a.k.a Gaussian kernel or RBF kernel, is given by:

$$k(x, x') = \sigma_z^2 exp[\frac{-|x - x'|^2}{2l^2}]$$

l controls the horizontal length scale over which the function varies, and σ_z^2 controls the vertical variation. RBF kernel will be used in the implementation later.

Problem 2. Show detailed calculation of the marginal likelihood in GP regression.

Solution Since $p(y, f|X) = p(y|f, X) \times p(f|X)$, we can apply logarithm on both sides:

$$\ln p(y, f|X) = \ln p(y|f, X) + \ln p(f|X)$$

Therefore we can derive follows: as $p(y|f, x) \sim \mathrm{N}(y|f, \sigma_N^2 I)$ and $p(f|x) \sim N(f|0, K)$,

$$\begin{aligned}
\ln p(y, f|X) &= \ln p(y|f, X) + \ln p(f|X) \\
&= -\frac{1}{2}\{(y - f)^T(\sigma^2 I_N)^{-1}(y - f)\} - \frac{1}{2}\{(f - 0)^T K^{-1}(f - 0)\} + const. \\
&= -\frac{1}{2}\{(\sigma^2)^{-1}y^T y - (\sigma^2)^{-1}f^T y - (\sigma^2)^{-1}y^T f + (\sigma^2)^{-1}f^T f\} - \frac{1}{2}(f^T K^{-1} f) + const. \\
&= -\frac{1}{2}\{f^T K^{-1} f + (\sigma^2)^{-1}y^T y - (\sigma^2)^{-1}f^T y - (\sigma^2)^{-1}y^T f + (\sigma^2)^{-1}f^T f\} + const. \\
&= -\frac{1}{2}\begin{pmatrix} \mathrm{f} \\ \mathrm{y} \end{pmatrix}^T \begin{pmatrix} \mathrm{K}^{-1} + (\sigma^2)^{-1}I_N & \text{-}(\sigma^2)^{-1}I_N \\ \text{-}(\sigma^2)^{-1}I_N & (\sigma^2)^{-1}I_N \end{pmatrix} \begin{pmatrix} \mathrm{f} \\ \mathrm{y} \end{pmatrix} + const.
\end{aligned}$$

Hence, if we write $p(T|X)$ instead of $p(f, y|X) = p(y, f|X)$, we can notice that $\ln p(f, y|X)$ can be represented as $\ln p(T|X) = -\frac{1}{2}T^T R^{-1} T + const.$, where T and R^{-1} is:

$$\mathrm{T} = \begin{pmatrix} \mathrm{f} \\ \mathrm{y} \end{pmatrix} \qquad \mathrm{R}^{-1} = \begin{pmatrix} \mathrm{K}^{-1} + (\sigma^2)^{-1}I_N & \text{-}(\sigma^2)^{-1}I_N \\ \text{-}(\sigma^2)^{-1}I_N & (\sigma^2)^{-1}I_N \end{pmatrix}$$

Then R can be derived by the formula in [4] *chpater 9.1.3 eq(399), eq(400)* as follows:

$$\mathrm{R} = \begin{pmatrix} \mathrm{K} & \mathrm{K} \\ \mathrm{K} & \mathrm{K}{+}(\sigma^2)I_N \end{pmatrix}$$

Now, we know that $p(T|X)$ follows a normal distribution of zero-mean i.e. $(0, 0)$ and covariance R. By the formula [3] *in chapter 4.3 eq(4.68)*, we can have $p(y|X)$ from jointly distribution $p(T|X) = p(f, y|X)$ as follows:

$$p(y|X) = \mathcal{N}(y|0, K + (\sigma^2)I_N)$$

Then, apply logarithm on the multivariate normal distribution of zero mean and covariance $K + (\sigma^2)I_N$ is that,

$$\begin{aligned}
\ln p(y|X) &= \ln\{(2\pi)^{-\frac{N}{2}}|K + (\sigma^2)I_N|^{-\frac{1}{2}} exp(-\frac{1}{2}y^T(K + (\sigma^2)I_N)^{-1}y)\} \\
&= -\frac{N}{2}\ln 2\pi - \frac{1}{2}\ln|K + (\sigma^2)I_N| - \frac{1}{2}y^T(K + (\sigma^2)I_N)^{-1}y
\end{aligned}$$

Problem 3. Do your own work to generate Fig. 15.3 (in Murphy's book), implementing your own GP regression code.

Solution From the above derivation we can conclude following pseudo-code:

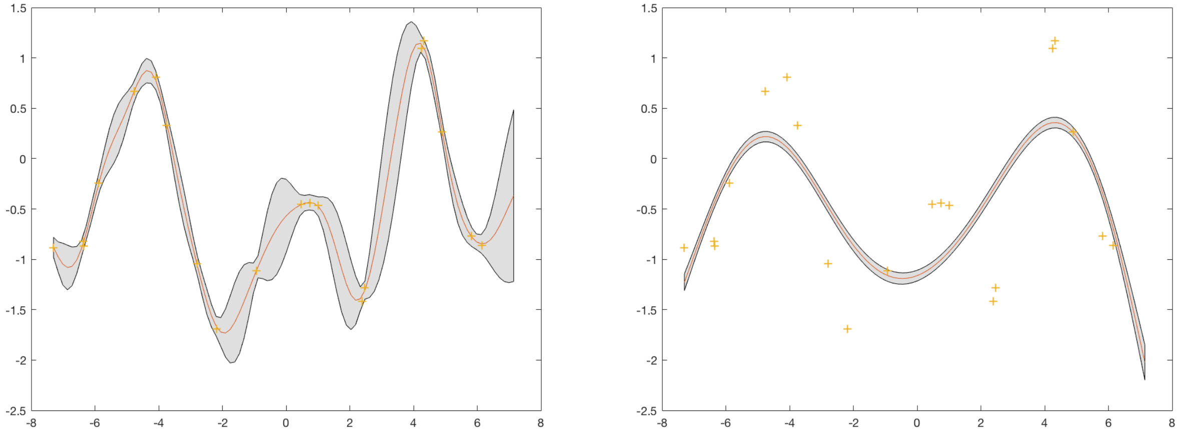
Algorithm 1: GP Regression

```
input  : Training Dataset $\mathcal{D} = (x_n, y_n)|n = 1, ..., N$, Test input $x_*$, covariance
         function $k(., .)$, and noise level $\sigma^2$
output: $\mathbb{E}(Z_{x_*}), var(Z_{x_*}), \log p(y|X)$
Create RBF function as k(x,x') = $\sigma_f^2 exp[\frac{-|x-x'|^2}{2l^2}]$
Compute$K = [k(x_i, x_j)]_{i,j \leq N}$ and $k_* = [k(x_1, x_*), ..., k(x_N, x_*)]^T$;
L = Cholesky$(K + \sigma^2 I)$;
$\alpha = (L^T)^{-1}(L^{-1}Y)$;
Compute predicted mean: $\mathbb{E}(Z_{x_*}) = k_*^T \alpha$;
$v = L^{-1}k_*$;
Compute predictive variance: $var(Z_{x_*}) = k(x_*, x_*) - v^T v$;
Compute the marginal log-likelihood:   $\log p(y|X) = -\frac{N}{2}\log 2\pi - \Sigma_n \log L_{n,n} - \frac{1}{2}y^T\alpha$;
```

For finding $p(Z_{x_*}|x_*, X, Z)$, we need to calculate K^{-1}. Matrix inversion is a classical problem, and can be very complicated for large matrices. Fortunately, K is a p.s.d so that we can use Cholesky decomposition, which breaks $K = LL^T = U^T U$. Hence, based on dataset and test data covariance matrix could be calculated easily. And $\log|K + \sigma^2|$ alos computed easily since it can be derived as $\log{(LL^T LL^T)}^{\frac{1}{2}} = \log tr(L)$. From the above code, A mean, covariance and likelihood value could be achieved. Since it was mentioned that a pseudo-code is sufficient, I wouldn't attach the full script.

Results

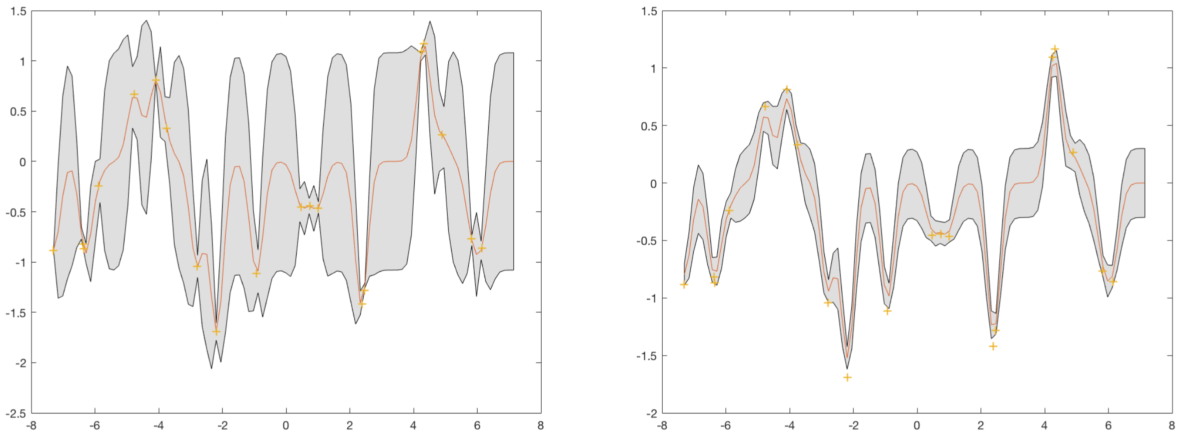
By the above, following results can be achieved upon given dataset.



(a) (1,1,0.1)

(b) (5,1,0.1)

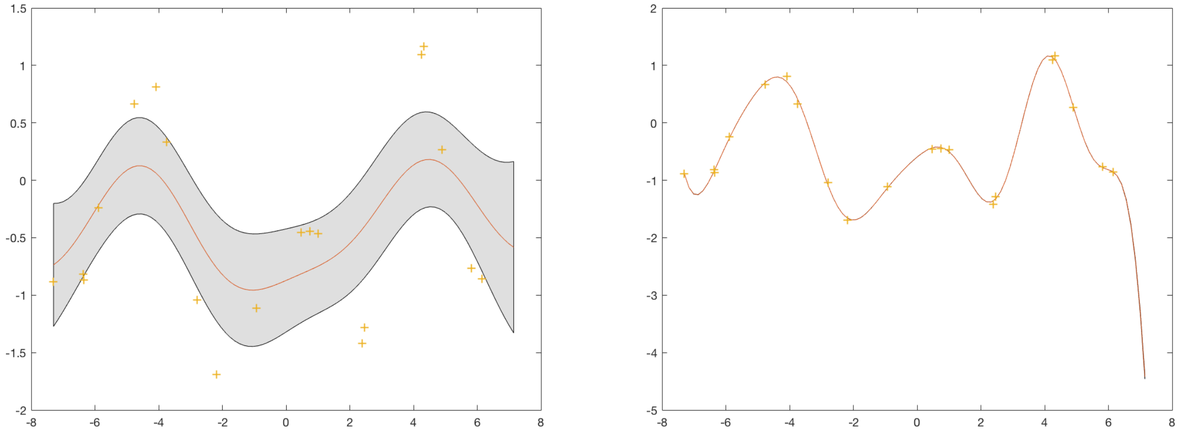
Figure 1: small vs large length



(a) (0.3,1.08,0.00005)

(b) (0.3,0.03,0.00005)

Figure 2: large vs small σ_f



(a) (3.0, 1.16, 0.89)

(b) (3.0, 1.16, 0.0009)

Figure 3: large vs small σ_y

Discussions

From [figure1], it can be noticed that length scale(l) makes whether function fast or not. I assume that length scale has kind of the role of window size so that if it is larger, then the kernel take in further data. This makes a function have smoother graph. As I changed the value 3 times larger than original one, (b) has much smoother but sluggish slope than (a)

It could be said that length scale concerns x-axis, on the contrary, σ_f concerns y-axis. σ_f determines variation of function from its mean. From [figure2] smaller σ_f has smaller area of variance in (b). Especially between two train data points, the uncertainty is much larger in (a).

Not like length scale and σ_f, σ_y is about data. It makes model concern how much data have its noise intrinsically (might be said stochastic error). Therefore, if it has larger value, then model try to have bigger variance to absorb the noise bound as (a). On the other hand, if it has smaller value like (b), then it shows that our model interpolate exactly on the given data point, however, the extrapolating part has very low confidence.

References

[1] Manjunath Krishnapur, epartment of Mathematics, Indian Institute of Science.
`http://math.iisc.ac.in/`$\sim$ *manju/GP/gaussianprocesses.html*.

[2] Kevin Murpy. *Machine Learning: A probabilistic Perspective.*

[3] Wolfram Mathworld.
`http://mathworld.wolfram.com/NormalSumDistribution.html`

[4] Kaare Brandt Petersen. *The Matrix Cookbook. Version: November 15, 2012.*