Probabilistic Machine Learning (CS772A), Spring 2023
Indian Institute of Technology Kanpur
Homework Assignment Number 2

Student Name: Shrey Mehta
Roll Number: 200580
Date: March 30, 2023

QUESTION 1

The Gamma pdf is given by $\text{Gamma}(x|a,b) = \frac{b^a}{\Gamma(a)}x^{a-1}e^{-bx}$ where a and b are the shape and rate parameters, respectively, and $\Gamma(a)$ denotes the gamma function.

- Using Laplace Transform, we have $p(|D) \approx \mathcal{N}(\theta_{MAP}, H^{-1})$, where H is the Hessian matrix.
  So to find the MAP estimate of the pdf, we have

$$\begin{aligned} x_{MAP} &= \arg\max_x[ln(p(x|a,b))] \\ &= \arg\max_x[ln(\frac{b^a}{\Gamma(a)}x^{a-1}e^{-bx})] \\ &= \arg\max_x[(a-1)ln(x) - bx \end{aligned}$$

  Differentaiting it, we get,

$$\frac{a-1}{x} - b = 0$$

  So, we get x = $\frac{a-1}{b}$ as the MAP estimate of the pdf.

$$\begin{aligned} H &= -\nabla^2(log(p(D|\theta))\ at\ \theta = \theta_{MAP} \\ &= -\frac{\partial^2}{\partial x^2}(ln((a-1)ln(x)) - bx) \\ &= -\frac{\partial}{\partial x}(\frac{a-1}{x} - b) \\ &= \frac{a-1}{x^2} \\ &= \frac{a-1}{(\frac{a-1}{b})^2} \\ &= \frac{b^2}{a-1} \end{aligned}$$

  So, by Laplace approximation, we have $\text{Gamma}(x|a,b) \approx \mathcal{N}(x|\frac{a-1}{b}, \frac{a-1}{b^2})$.
  The mean of the Gamma function is $\frac{a}{b}$ and variance is $\frac{a}{b^2}$. So, the Gaussian whose mean and variance are equal to the mean and variance of the Gamma function is $\mathcal{N}(\frac{a}{b}, \frac{a}{b^2})$.
  This is roughly equal to the Gaussian obtained from the Laplace approximation when

$$\frac{a}{b} \approx \frac{a-1}{b}$$

  which happens when b $\to \infty$

- From the Laplace approximation, we get $\Gamma(a)$ at MAP estimate of x

$$\frac{b}{\sqrt{(2\pi(a-1))}} e^{-\frac{(x-\frac{a-1}{b})^2}{\frac{a-1}{b^2}}} = \frac{b^a}{\Gamma(a)} x^{a-1} e^{-bx}$$

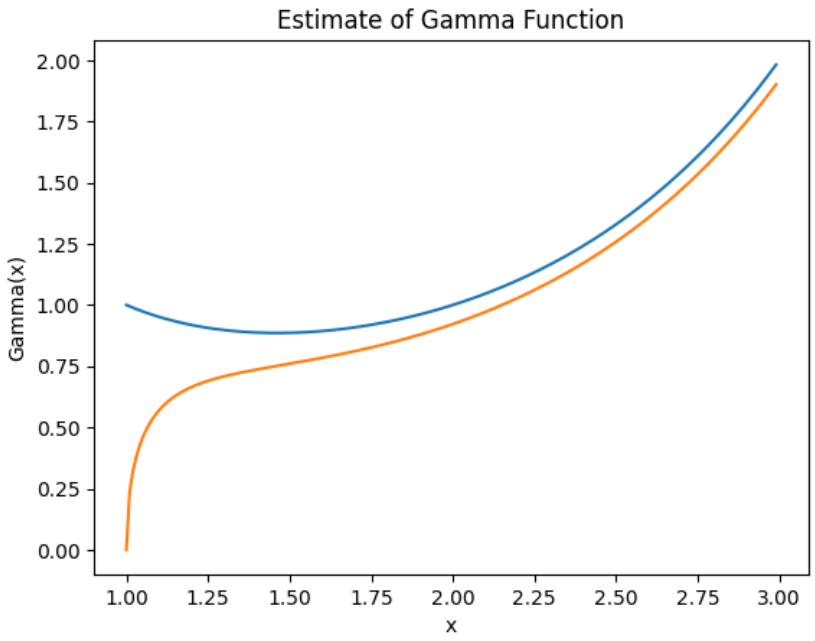
Getting the value of $\Gamma(a)$ from above, we get

$$\Gamma(a) = \sqrt{2\pi(a-1)} b^{a-1} x^{a-1} e^{\frac{-(bx-(a-1))^2}{2*(a-1)} - bx}$$

Now, putting the MAP estimate of x in the above equation, we get

$$\Gamma(a) = \sqrt{2\pi(a-1)} (a-1)^{a-1} e^{-(a-1)}$$

Below is the plot for the $\Gamma(x)$ plotted using numpy (blue) and the one approximated using the laplace approximation (orange) plotted from x = 1 to x = 3 (Code for plotting attached along with the submission).



Probabilistic Machine Learning (CS772A), Spring 2023
Indian Institute of Technology Kanpur
Homework Assignment Number 2

Student Name: Shrey Mehta
Roll Number: 200580
Date: March 30, 2023

QUESTION 2

We have N scalar observations $x_1, x_2, ..., x_N$ drawn i.i.d. from a Gaussian $\mathcal{N}(x|\mu, \beta^{-1})$. The mean μ has a Gaussian prior $\mathcal{N}(\mu|\mu_0, s_0)$ where μ_0 is the mean and s_0 is the variance of the Gaussian prior. The precision β has a gamma prior Gamma$(\beta|a, b)$ where a and b are the shape and rate parameters, respectively, of the gamma prior. So, using the idea of local conjugacy, we have the the conditional posteriors:

$$\begin{aligned} p(\mu|x, \beta^{-1}) &\propto p(x|\mu, \beta^{-1})p(\mu) \\ &\propto (\prod_{n=1}^{N} \mathcal{N}(x_i|\mu, \beta^{-1}))\mathcal{N}(\mu|\mu_0, s_0) \end{aligned} \tag{1}$$

$$\begin{aligned} p(\beta|x, \mu) &\propto p(x|\mu, \beta^{-1})p(\mu) \\ &\propto (\prod_{n=1}^{N} \mathcal{N}(x_i|\mu, \beta^{-1}))\ \text{Gamma}(\beta|a, b) \end{aligned} \tag{2}$$

So, from equation (1), using the result from the slides, we have

$$\begin{aligned} p(\mu|x, \beta^{-1}) &\propto (\prod_{n=1}^{N} \mathcal{N}(x_i|\mu, \beta^{-1}))\mathcal{N}(\mu|\mu_0, s_0) \\ &\propto \mathcal{N}(\mu_N, \sigma_N^2) \end{aligned}$$

where,

$$\mu_N = \frac{\mu_0}{N\beta s_0} + \frac{N\beta s_0}{N\beta s_0 + 1}\bar{x} \text{ where } \bar{x} = \frac{\Sigma_{n=1}^{N} x_i}{N}$$
$$\frac{1}{\sigma_N^2} = \frac{1}{s_0} + N\beta$$

From equation (2), we have

$$\begin{aligned} p(\beta|x, \mu) &\propto (\prod_{n=1}^{N} \mathcal{N}(x_i|\mu, \beta^{-1}))\ \text{Gamma}(\beta|a, b) \\ &\propto \text{Gamma}(\beta|a^{'}, b^{'}) \end{aligned}$$

where $a^{'} = a + \frac{N}{2}$ and $b^{'} = b + \frac{\Sigma_{n=1}^{N}(x_n - \mu)^2}{2}$ using the result from the slides.
Now, we can estimate the joint posterior of μ and β using Gibb's sampling which works as:

1. Initialize $\mu^{(0)}$ to μ_0.

2. For s = 1, 2 ..... S :

   - Draw a random sample for β as $\beta^{(s)} \sim p(\beta|\mathbf{x}, \mu^{(s-1)})$
   - Draw a random sample for μ as $\mu^{(s)} \sim p(\mu|\mathbf{x}, \beta^{(s)})$

These S samples $(\mu^{(s)}, \beta^{(s)})_{s=1}^{S}$ represent the joint posterior $p(\beta, \mu|\mathbf{x})$.

Probabilistic Machine Learning (CS772A), Spring 2023
Indian Institute of Technology Kanpur
Homework Assignment Number 2

Student Name: Shrey Mehta
Roll Number: 200580
Date: March 30, 2023

QUESTION 3

We have the following distributions given:

$$(\mathbf{y}|\mathbf{X}, \mathbf{w}, \beta) \sim \prod_{n=1}^{N} \mathcal{N}(y_n|\mathbf{w}^T\mathbf{x}_n, \beta^{-1}) = \mathcal{N}(\mathbf{y}|\mathbf{X}\mathbf{w}, \beta^{-1}\mathbf{I}_D)$$

as described in the lectures.

$$(\mathbf{w}|\lambda) \sim \mathcal{N}(\mathbf{w}|\mathbf{0}, \lambda^{-1}\mathbf{I}_D)$$

So, we get $p(\mathbf{w}|\mathbf{y}, \mathbf{X}, \lambda, \beta) = \mathcal{N}(\mathbf{y}|\mathbf{X}\mathbf{w}, \beta^{-1}\mathbf{I}_N)\mathcal{N}(\mathbf{w}|\mathbf{0}, \lambda^{-1}\mathbf{I}_D) = \mathcal{N}(\mathbf{w}|\boldsymbol{\mu}_N, \boldsymbol{\Sigma}_N)$.

Here $\boldsymbol{\mu}_N = (\mathbf{X^T X} + \frac{\lambda}{\beta}\mathbf{I}_D)^{-1}\mathbf{X^T y}$ and $\boldsymbol{\Sigma}_N = (\beta\mathbf{X^T X} + \lambda\mathbf{I}_D)^{-1}$.

Using these distributions, we can find the complete log likelihood (CLL) to be $log(p(\mathbf{y}, \mathbf{w}|\mathbf{X}, \beta, \lambda))$.

Now, $p(\mathbf{y}, \mathbf{w}|\mathbf{X}, \beta, \lambda) = p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \beta)p(\mathbf{w}|\lambda)$. So, we have

$$\begin{aligned} log(p(\mathbf{y}, \mathbf{w}|\mathbf{X}, \beta, \lambda)) &= log(p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \beta)) + log(p(\mathbf{w}|\lambda)) \\ &= \frac{1}{2}(Nlog(\beta) + Dlog(\lambda) - \beta(\mathbf{y} - \mathbf{X}\mathbf{w})^T(\mathbf{y} - \mathbf{X}\mathbf{w}) - \lambda\mathbf{w^T w} - (N + D)log(2 * \pi)) \end{aligned}$$

This value of the complete log likelihood is calculated using the fact that the value of $\mathbf{w}$ is observed, so we have to use the EM algorithm to alternately model the global and the local parameters of the model. The EM algorithm has 2 steps :

- **Expectation Step**
  We assume some initial values for the hyperparameters λ and β. From the lectures, it is known that we can calculate the expectation with respect to the posterior of the latent variable $\mathbf{w}$. We are treating $\mathbf{w}$ as the latent variable and λ and β as the hyperparameters in this case. So, in this step, we find the expectation of w to be

$$\begin{aligned} E[\mathbf{w}] &= \int_{w \in R^D} p(\mathbf{w}|\mathbf{y}, \mathbf{X}, \lambda, \beta)\mathbf{w}d\mathbf{w} \\ &= \int_{\mathbf{w} \in R^D} \mathbf{w}\mathcal{N}(\mathbf{w}|\boldsymbol{\mu}_N, \boldsymbol{\Sigma}_N)d\mathbf{w} \end{aligned}$$

  We know that the expectation of the normal distribution is equal to the mean of the distribution, so the above expectation is equal to $\boldsymbol{\mu}_N$.
  We now need to maximise the expectation of the complete log likelihood which is

given by

$$E[CLL] = E[\frac{1}{2}(Nlog(\beta) + Dlog(\lambda) - \beta(\mathbf{y} - \mathbf{X}\mathbf{w})^T(\mathbf{y} - \mathbf{X}\mathbf{w}) - \lambda\mathbf{w}^\mathbf{T}\mathbf{w} - (N+D)log(2*\pi))]$$
$$= \frac{1}{2}(Nlog(\beta) + Dlog(\lambda) - (N+D)log(2*\pi)) - \beta(\mathbf{y}^\mathbf{T}\mathbf{y} - \mathbf{y}^\mathbf{T}\mathbf{X}E[\mathbf{w}] - E[\mathbf{w}^\mathbf{T}]\mathbf{X}^\mathbf{T}\mathbf{y})$$
$$+ \beta(E[\mathbf{w}^\mathbf{T}(\mathbf{X}^\mathbf{T}\mathbf{X} + \frac{\lambda}{\beta}\mathbf{I}_D)\mathbf{w}]))$$

Now, as given in the slides we replace the values of the latent variable $\mathbf{w}$ as follows:

$$E[\mathbf{w}^\mathbf{T}] = E[\mathbf{w}]^T = \boldsymbol{\mu}_N^T$$

$$E[\mathbf{w}\mathbf{w}^\mathbf{T}] = E[\mathbf{w}]E[\mathbf{w}]^T + Cov(w) = \boldsymbol{\mu}_N\boldsymbol{\mu}_N^T + \boldsymbol{\Sigma}_N$$
$$E[\mathbf{w}^\mathbf{T}R\mathbf{w}] = Trace(R * E[\mathbf{w}\mathbf{w}^\mathbf{T}]) = Trace(R(\boldsymbol{\Sigma} + \boldsymbol{\mu_N}\boldsymbol{\mu_N^T}))$$

So, putting these values in the E[CLL], we get,

$$E[CLL] = \frac{1}{2}(Nlog(\beta) + Dlog(\lambda) - (N+D)log(2*\pi) - \beta(\mathbf{y}^\mathbf{T}\mathbf{y} - \mathbf{y}^\mathbf{T}\mathbf{X}\boldsymbol{\mu_N} - \boldsymbol{\mu}_\mathbf{N}^\mathbf{T}\mathbf{X}^\mathbf{T}\mathbf{y})$$
$$+ \beta(Trace(\mathbf{X}^\mathbf{T}\mathbf{X} + \frac{\lambda}{\beta}\mathbf{I}_D)(\boldsymbol{\mu_N}\boldsymbol{\mu_N^T} + \boldsymbol{\Sigma}_N)))$$

- **Maximization Step**
  We are required to get the MLE estimates of the expectation obtained above wrt to the hyperparameters. So,
  - MLE Estimate of β
    We maximize the E[CLL] wrt to β:

    $\beta_{MLE} = \arg\max_\beta E[CLL]$ Solving this, we get:

    $$\beta_{MLE}^{-1} = \frac{1}{N}(\mathbf{y}^\mathbf{T}\mathbf{y} - 2\boldsymbol{\mu}_\mathbf{N}^\mathbf{T}\boldsymbol{X}^\boldsymbol{T}\boldsymbol{y} + Trace((\mathbf{X}^\mathbf{T}\mathbf{X})(\boldsymbol{\mu_N}\boldsymbol{\mu_N^T} + \boldsymbol{\Sigma}_N)))$$
  - MLE Estimate of λ
    We maximize the E[CLL] wrt to β:

    $\lambda_{MLE} = \arg\max_\lambda E[CLL]$ Solving this, we get:

    $$\lambda_{MLE}^{-1} = \frac{Trace((\boldsymbol{\mu_N}\boldsymbol{\mu}_\mathbf{N}^\mathbf{T} + \boldsymbol{\Sigma_N}))}{D}$$

    where D is the dimension of the vector space.

The EM algorithm iteratively repeats the E and M steps until convergence or a stopping criterion is met. The convergence criterion can be set to, for example, a maximum number of iterations, a minimum change in the parameter estimate, or a maximum change in the CLL.

Overall Sketch of the EM Algorithm

Steps:

1. Initialise the value of λ and β to λ_0 and β_0, respectively
2. Repeat the following steps until convergence for steps t = 1 to T:

   - **E Step**
     Given $\lambda^{(t-1)}$ and $\beta^{(t-1)}$, compute the posterior distribution:

     $$p(\mathbf{w}^{(t-1)}|\mathbf{y}, \mathbf{X}, \lambda^{(t-1)}, \beta^{(t-1)}) = \mathcal{N}(\mathbf{w}|\boldsymbol{\mu}_N, \boldsymbol{\Sigma}_N)$$

     Here $\boldsymbol{\mu}_N = (\mathbf{X^T X} + \frac{\lambda^{(t-1)}}{\beta^{(t-1)}}\mathbf{I}_D)^{-1}\mathbf{X^T y}$ and $\boldsymbol{\Sigma}_N = (\beta^{(t-1)}\mathbf{X^T X} + \lambda^{(t-1)}\mathbf{I}_D)^{-1}$.
     Then, compute the expectations given by:

     $$E[\mathbf{w^T}] = E[\mathbf{w}]^T = \boldsymbol{\mu}_N^T$$

     $$E[\mathbf{w w^T}] = E[\mathbf{w}]E[\mathbf{w}]^T + Cov(w) = \boldsymbol{\mu}_N\boldsymbol{\mu}_N^T + \boldsymbol{\Sigma}_N$$
     $$E[\mathbf{w^T w}] = Trace(E[\mathbf{w w^T}]) = Trace(\boldsymbol{\Sigma} + \boldsymbol{\mu_N}\boldsymbol{\mu_N^T})$$

   - **M Step**
     Compute the MLE solution of λ and β:

     $$(\lambda^{(t)})^{-1} = \frac{Trace(\boldsymbol{\mu_N}\boldsymbol{\mu_N^T} + \boldsymbol{\Sigma_N})}{D}$$

     $$(\beta^{(t)})^{-1} = \frac{1}{N}(\mathbf{y^T y} - 2\boldsymbol{\mu_N^T}\boldsymbol{X^T}\boldsymbol{y} + Trace((\mathbf{X^T X})(\boldsymbol{\mu_N}\boldsymbol{\mu_N^T} + \boldsymbol{\Sigma}_N))$$

   If not converged yet, go back to **E Step**.

Probabilistic Machine Learning (CS772A), Spring 2023
Indian Institute of Technology Kanpur
Homework Assignment Number 2

Student Name: Shrey Mehta
Roll Number: 200580
Date: March 30, 2023

QUESTION 4

We are given a binary classification with training data $(x_n, y_n)_{n=1}^N$, with $x_n \in R^D$ and $y_n \in \{0, 1\}$.
Each binary label to be generated as $y_n = \mathbb{I}[z_n > 0]$. We are given that:

$$p(z_n|\mathbf{w}, \mathbf{x_n}) = \mathcal{N}(\mathbf{w^T x}, 1)$$

where $\mathbf{w} \in \mathbb{R}^D$ and $\mathbb{I}[.]$ returns 1 if the condition is true, else it returns 0.
It is given that $\mathbf{z_n}$ is a Gaussian latent variable, so we assume a Gaussian prior on w of the form $\mathcal{N}(\mathbf{0}, \boldsymbol{\beta})$.
Model parameter is $\mathbf{w}$ and the goal is point estimation for $\mathbf{w}$. So, we takes two cases of $y_n = 1$ and $y_n = 0$ separately. Now, it is clear that

$$p(z_n|y_n = 1, \mathbf{x_n}, \mathbf{w}) = p(z_n|z_n > 0, \mathbf{x_n}, \mathbf{w})$$

Now, this is similar to the **Truncated Gaussian Distribution**, so we can write this as

$$p(z_n|y_n = 1, \mathbf{x_n}, \mathbf{w}) = \mathbb{I}(z_n > 0)\frac{\mathcal{N}(z_n|\mathbf{w^T x_n}, 1)}{1 - \Phi(-\mathbf{w^T x_n})}$$

Here $\Phi(.)$ is the cumulative distribution function of the standard Gaussian distribution.
So we can have the probability function for the other case as well:

$$p(z_n|y_n = 0, \mathbf{x_n}, \mathbf{w}) = \mathbb{I}(z_n < 0)\frac{\mathcal{N}(z_n|\mathbf{w^T x_n}, 1)}{\Phi(-\mathbf{w^T x_n})}$$

Combining these two equations, we get

$$p(z_n|y_n, \mathbf{x_n}, \mathbf{w}) = [y_n\frac{\mathbb{I}(z_n > 0)}{1 - \Phi(-\mathbf{w^T x_n})} + (1 - y_n)\frac{\mathbb{I}(z_n < 0)}{\Phi(-\mathbf{w^T x_n})}]\mathcal{N}(z_n|\mathbf{w^T x_n}, 1)$$

The posterior distribution is given by

$$\begin{aligned}
p(\mathbf{y}, \mathbf{z}|\mathbf{X}, \mathbf{w}) &= p(\mathbf{z}|\mathbf{y}, \mathbf{X}, \mathbf{w})p(\mathbf{y}|\mathbf{X}, \mathbf{w}) \\
&= \prod_{n=1}^{N}(p(z_n|y_n, \mathbf{x}_n, \mathbf{w})p(y_n|\mathbf{x}_n, \mathbf{w})) \\
&= [y_n\frac{\mathbb{I}(z_n > 0)}{1 - \Phi(-\mathbf{w^T x_n})} + (1 - y_n)\frac{\mathbb{I}(z_n < 0)}{\Phi(-\mathbf{w^T x_n})}]\mathcal{N}(z_n|\mathbf{w^T x_n}, 1) \\
&\qquad [y_n(1 - \Phi(-\mathbf{w^T x_n})) + (1 - y_n)(\Phi(-\mathbf{w^T x_N}))] \\
&= \mathcal{N}(z_n|\mathbf{w^T x_n}, 1)(y_n\mathbb{I}(z_n > 0) + (1 - y_n)\mathbb{I}(z_n < 0))
\end{aligned}$$

The Complete Data Log Likelihood (CLL) for the given posterior distribution is given by

$$CLL = \Sigma_{n=1}^{N}[-\frac{1}{2}(log(2\pi)) - \frac{1}{2}(z_n - \mathbf{w^T x}_n)^2 + y_n log(\mathbb{I}(z_n > 0)) + (1 - y_n) log(\mathbb{I}(z_n < 0))]$$

Now, we use the EM algorithm for this binary classification model.

- **Expectation Step**
  Here, we calculate the expectation of the CLL, which is given by:

$$\begin{aligned} E[CLL] = \Sigma_{n=1}^{N}[-\frac{1}{2}(log(2\pi)) - \frac{1}{2}(E[z_n^2] + \mathbf{w^T x}_n^2 - 2\mathbf{w^T x_n} E[z_n]) \\ + y_n E[log(\mathbb{I}(z_n > 0))] + (1 - y_n) E[log(\mathbb{I}(z_n < 0))]] \\ = \Sigma_{n=1}^{N}[-\frac{1}{2}(log(2\pi)) - \frac{1}{2}(E[z_n^2] + (\mathbf{w^T x}_n)^2 - 2\mathbf{w^T x_n} E[z_n])] \end{aligned}$$

  The last expectation of CLL is obtained by separately computing the cases for y_n = 0 and y_n = 1. Now, we need to compute the above expectations in order to use them in the maximisation step. We use the value of $\mathbf{w}$ obtained in step (t-1) to compute these expectations in step t. Let $\mathbf{w}$ in step (t-1) be given by $\mathbf{w}^{(t-1)}$.

$$E[z_n] \text{ (at y = 1)} = \mathbf{w}^{(t-1)^T} + \frac{\phi(\mathbf{w}^{(t-1)^T}\mathbf{x}_n)}{1 - \Phi(-\mathbf{w}^{(t-1)^T})}$$

$$E[z_n] \text{ (at y = 0)} = \mathbf{w}^{(t-1)^T} + \frac{\phi(\mathbf{w}^{(t-1)^T}\mathbf{x}_n)}{\Phi(-\mathbf{w}^{(t-1)^T})}$$

  where $\phi(.)$ is the standard pdf of the Gaussian distribution fucntion.
  So, the total expectation comes out to be

$$E[z_n] = \mathbf{w}^{(t-1)^T} + \phi(\mathbf{w}^{(t-1)^T}\mathbf{x}_n) \left(\frac{1}{1 - \Phi(-\mathbf{w}^{(t-1)^T})}\right)^{y_n} \left(\frac{-1}{\Phi(-\mathbf{w}^{(t-1)^T})}\right)^{1-y_n}$$

  In the maximization step, we will maximise the E[CLL] wrt to w, so we don't require to calculate E$[z_n^2]$ and it can be treated as a constant.

- **Maximization Step**
  We need to perform MLE estimate for w by maximizing the E[CLL]. So,

$$\hat{\mathbf{w}}_{MLE} = \arg\max_{\mathbf{w}} E[CLL]$$

  Differentiating the expectation wrt to $\mathbf{w}$ and equating it to 0, we get

$$\Sigma_{n=1}^{N}(\mathbf{w}^T\mathbf{x}_n)\mathbf{x}_n = E[z_n]\mathbf{x}_n$$

$$\mathbf{X^T X w} = \mathbf{X^T} E[\mathbf{z}]$$

  So, we get, $\hat{\mathbf{w}}_{MLE} = (\mathbf{X^T X}^{-1})\mathbf{X^T} E[\mathbf{x}]$ where E$[\mathbf{z}] = [E[\mathbf{z_1}], E[\mathbf{z_2}]....E[\mathbf{z_N}]]^T$
  The EM algorithm iteratively repeats the E and M steps until convergence or a stopping criterion is met. The convergence criterion can be set to, for example, a maximum number of iterations, a minimum change in the parameter estimate, or a maximum change in the CLL.

Overall Sketch of the EM Algorithm

Steps:

1. Initialise the value of $\mathbf{w}$ to $\mathbf{w_0}$
2. Repeat the following steps until convergence for steps t = 1 to T:

- **E Step**
  Given $\mathbf{w}^{(t-1)}$, compute the posterior distribution:

$$p(z_n|y_n, \mathbf{x_n}, \mathbf{w}^{(t-1)}) = \left[y_n \frac{\mathbb{I}(z_n > 0)}{1 - \Phi(-\mathbf{w}^{(t-1)^T}\mathbf{x_n})} + (1 - y_n) \frac{\mathbb{I}(z_n < 0)}{\Phi(-\mathbf{w}^{(t-1)^T}\mathbf{x_n})} \right] \mathcal{N}(\mathbf{w}^{(t-1)^T}\mathbf{x_n}, 1)$$

  Then, compute the expectation of the z_n given by:

$$E[z_n] = \mathbf{w}^{(t-1)^T} + \phi(\mathbf{w}^{(t-1)^T}\mathbf{x}_n) \left(\frac{1}{1 - \Phi(-\mathbf{w}^{(t-1)^T})} \right)^{y_n} \left(\frac{-1}{\Phi(-\mathbf{w}^{(t-1)^T})} \right)^{1-y_n}$$

- **M Step**
  Compute the MLE solution of $\mathbf{w}$:

$$\hat{\mathbf{w}}^{(t)} = (\mathbf{X^T X}^{-1})\mathbf{X^T} E[\mathbf{x}] \text{ where } E[\mathbf{z}] = [E[\mathbf{z_1}], E[\mathbf{z_2}]....E[\mathbf{z_N}]]^T$$

If not converged yet, go back to **E Step**.

Probabilistic Machine Learning (CS772A), Spring 2023
Indian Institute of Technology Kanpur
Homework Assignment Number 2

Student Name: Shrey Mehta
Roll Number: 200580
Date: March 30, 2023

QUESTION 5

- **Part1 : GP Posterior**

We are given the training input $(\mathbf{X}, \mathbf{y}) = \{\mathbf{x}_n, y_n\}_{n=1}^N$. Assuming the Guassian prior to be of the form $\mathrm{p}(\mathbf{f}) = \mathcal{N}(\mathbf{0}, \mathbf{K})$, where $\mathbf{K}$ is a N X Kernel matrix with $K_{nm} = \kappa(\mathbf{x_n}, \mathbf{x_m})$ and the likelihood model of the form $p(y_n|x_n, f) = \mathcal{N}(y_N|f(\mathbf{x_n}), \sigma^2)$ where $f \sim GP(0, \kappa)$. The GP posterior $p(\mathbf{f}|\mathbf{y})$ is of the form

$$p(\mathbf{f}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f})}{p(\mathbf{y})}$$

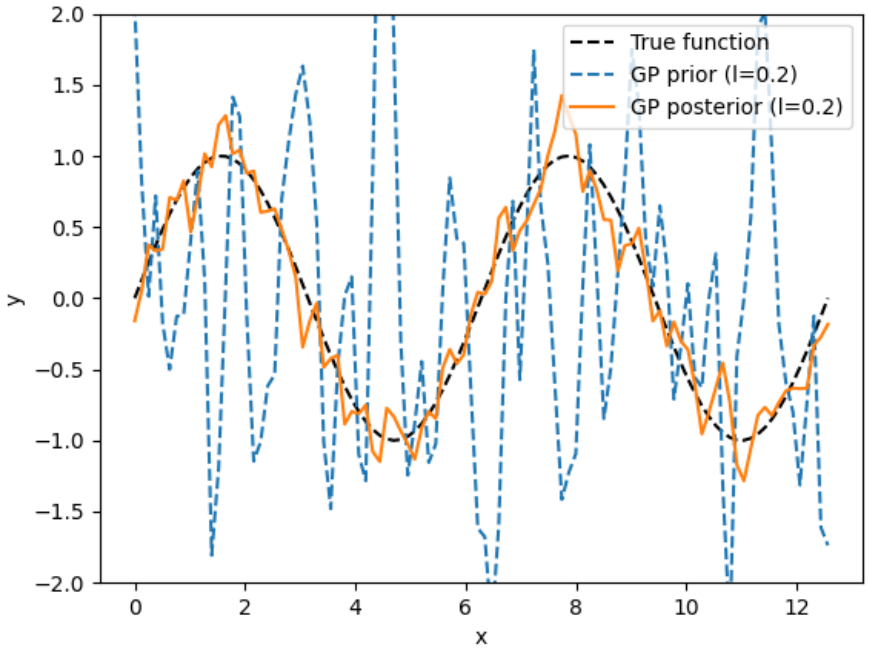
Here $p(\mathbf{y})$ is independent of $\mathbf{f}$, so the posterior becomes

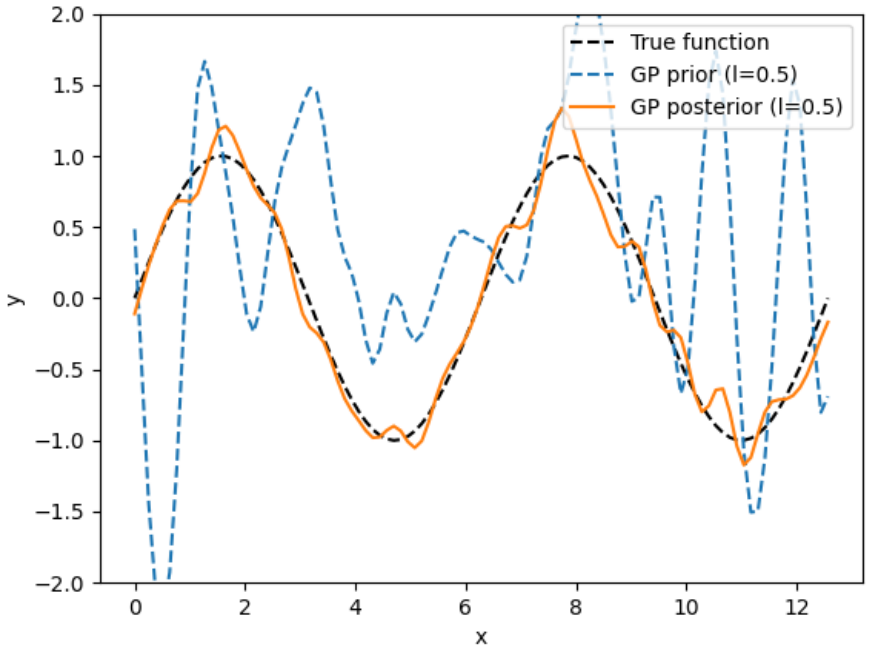
$$\begin{aligned} p(\mathbf{f}|\mathbf{y}) &\propto p(\mathbf{y}|\mathbf{f})p(\mathbf{f}) \\ &\propto (\prod_{n=1}^{N} \mathcal{N}(y_N|f(\mathbf{x_n}), \sigma^2))\mathcal{N}(\mathbf{0}, \mathbf{K}) \\ &\propto \mathcal{N}(\mathbf{y}|\mathbf{f}, \sigma^2\mathbf{I}_N)\mathcal{N}(\mathbf{0}, \mathbf{K}) \\ &\propto \mathcal{N}(\mathbf{f}|\boldsymbol{\mu}_N, \boldsymbol{\Sigma}_N) \text{ where } \boldsymbol{\Sigma}_N = \sigma^2(\sigma^2\mathbf{I}_N + \mathbf{K})^{-1}\mathbf{K} \text{ and } \boldsymbol{\mu}_N = \mathbf{K}(\sigma^2\mathbf{I}_N + \mathbf{K})^{-1}\mathbf{y} \end{aligned}$$

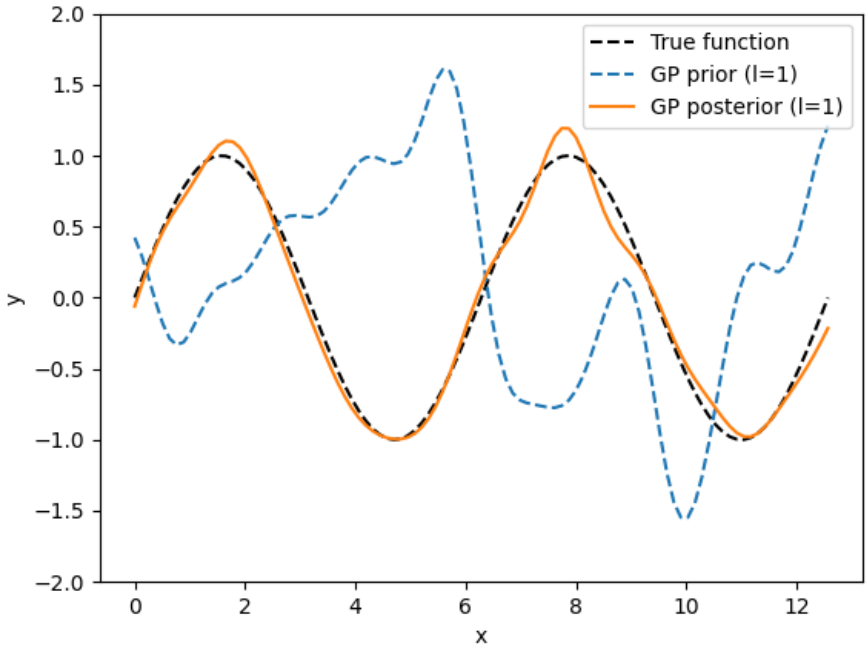
So, the GP Posterior comes out to be of the form

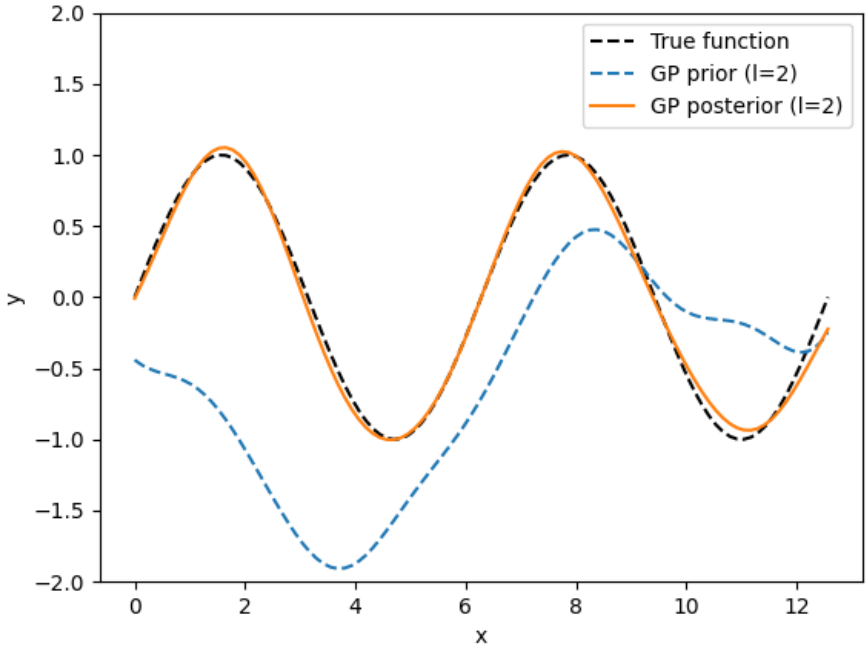
$$p(\mathbf{f}|\mathbf{y}) = \mathcal{N}(\boldsymbol{\mu}_N, \boldsymbol{\Sigma}_N)$$

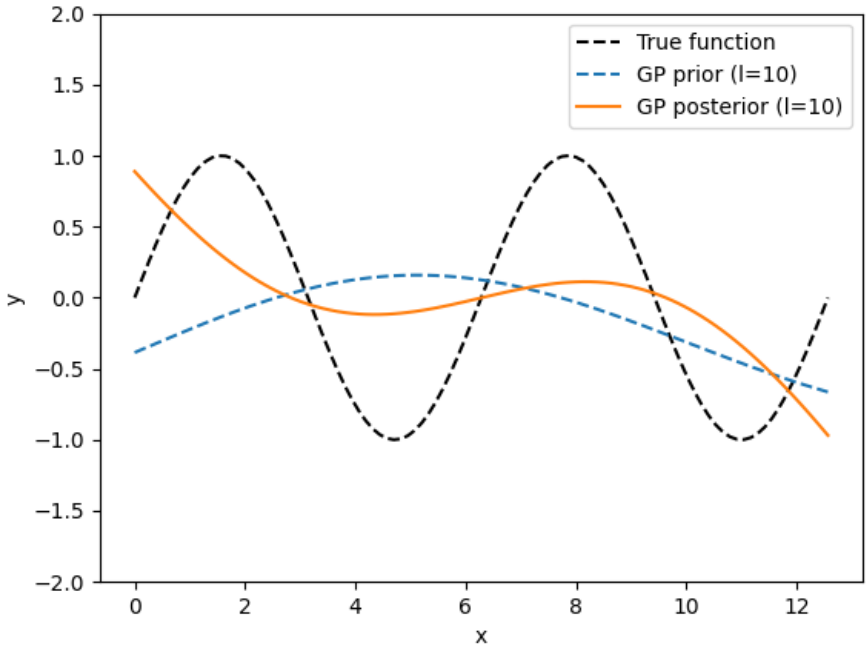
Part 2: Visualizing GP Priors and Posteriors for Regression
The following plots have been obtained on plotting for the given 5 values of l:






True function
GP prior (l=1)
GP posterior (l=1)
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
y
0
2
4
6
8
10
12
x


True function
GP prior (l=2)
GP posterior (l=2)
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
y
0
2
4
6
8
10
12
x



The difference between the plots generated using different values of the parameter l is mainly in the smoothness and amplitude of the GP prior and posterior functions, as well as in the uncertainty estimates. Here are some specific observations:

- For smaller values of l, the GP functions tend to be more wiggly and have higher frequency variations. This is because the SE kernel assigns a high covariance between inputs that are close to each other, which leads to a high correlation between function values at nearby input points. As a result, the GP samples tend to oscillate rapidly between data points.
- Conversely, for larger values of l, the GP functions tend to be smoother and have lower frequency variations. This is because the SE kernel assigns a low covariance between inputs that are far apart from each other, which leads to a lower correlation between function values at distant input points. As a result, the GP samples tend to vary more slowly and smoothly between data points.
- The GP posterior functions tend to be smoother than the GP prior functions, because they are conditioned on the observed data and thus have reduced uncertainty. The degree of smoothing depends on the noise level parameter sigma and the uncertainty in the data.
- The true function is a smooth curve with a sinusoidal shape, and it lies within the range of the GP prior and posterior functions for all values of l. However, the GP functions have more uncertainty than the true function, especially in regions with few or no data points. The uncertainty tends to be higher for smaller values of l, because the GP functions have more rapid variations and oscillations in those regions.